FINAL REPORT

STUDY OF THE MULTI-LANGUAGE PROBLEM IN COINS

VOLUME I – RECOMMENDED SOLUTIONS

Contract No. DAAB03-75-C-0105

Aug 1974 – May 1975

SDR 192

PREFACE

This document was prepared under Contract Number DAABO3-75-C-0105 by Logicon, Inc., 4010 Sorrento Valley Boulevard, San Diego, California 92138. This report reflects the combined efforts of a project team from the Applied Data System Department of the San Diego Division.

ABSTRACT

The Community Online Intelligence Network System (COINS) began in 1965 as an experiment and was established as a permanent facility for the intelligence community in 1973. Two major problems have hampered effective use of COINS since its inception: lack of interactive capability in the network and the need for users to know multiple query languages to access COINS files. The first problem is being resolved through the installation of COINS II which will provide excellent interactive capabilities to its users. The second problem, the COINS multi-language problem, continues to worsen with the planned and potential addition of one new node per year — each with its own query language.

The approach taken to provide a solution to the COINS multi-language problem consisted of the following tasks:

Developing a concise description of the COINS multi-language problem.

Defining requirements to be satisfied by solutions to the COINS multi-language problem.

Analyzing current and potential COINS data base management systems (DBMS) and their associated query languages to determine characteristics and capabilities.

Identifying and analyzing advanced techniques with potential for solution of the multi-language problem.

Identifying and describing potential solutions for the COINS multi-language problem.

Evaluating each potential solution with respect to COINS II requirements and selecting a recommended solution for the COINS multi-language problem.

Based upon the results of this study, the following approach is recommended as the best solution for the COINS multi-language problem:

a. Construct and install uniform terminal processors at each node.

b. Select and install the software version(s) of a standard DBMS at each host node.

c. Develop a user data language processor to translate user requests from the COINS user data language into standard DBMS commands and the standard DBMS responses into COINS user data language.

d. Upgrade the uniform terminal processors with the COINS user data language processor software thereby providing a uniform user-system-data base interface at each node throughout COINS.

e. Initiate development of a dedicated hardware version of the standard DBMS chosen for COINS.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

SECTION 1

INTRODUCTION

The Community Online Intelligence Network System (COINS) began in 1965 as an experiment and was established as a permanent facility for the intelligence community in 1973. Two major problems have hampered effective use of COINS since its inception: lack of interactive capability in the network and the need for users to know multiple query languages to access COINS files. The first problem is being resolved through the installation of COINS II which will provide excellent interactive capabilities to its users. The second problem, the COINS multi-language problem, continues to worsen with the planned and potential addition of one new node per year — each with its own query language.

The COINS PMO contracted with Logicon to carry out a study of the COINS multi-language problem. This study consisted of the following tasks:

a. Develop a concise description of the COINS multi-language problem.

b. Define requirements to be satisfied by solutions to the COINS multi-language problem.

c. Analyze current and potential COINS data base management systems (DBMS) and their associated query languages to determine characteristics and capabilities.

d. Identify and analyze advanced techniques with potential for solution of the multi-language problem.

e. Identify and describe potential solutions for the COINS multi-language problem.

f. Evaluate each potential solution with respect to COINS II requirements and select a recommended solution for the COINS multi-language problem.

g. Prepare a final report which describes the results of the study and the recommended solution, and produce a set of 35 mm slides for presenting the recommended solution to appropriate intelligence community personnel.

This document is Volume I of the COINS Multi-language Study Final Report. Volume I consists of seven additional sections. Section 2 provides a concise description of the COINS multi-language problem. Section 3 describes the requirements established for evaluating solutions to the multi-language problem. Section 4 contains summary descriptions of current and potential COINS DBMS and their associated query languages. (Volume II contains feature analysis for each of these DBMSs). Section 5 describes advanced technologies which were identified and analyzed for possible application to the COINS problem. Section 6 contains descriptions of each of the potential solutions identified during the study. Section 7 describes the evaluation of these solutions with respect to the previously developed requirements. Section 8 describes Logicon's recommended solution for resolving the COINS multi-language problem. This description consists of the selection rationale, tasks, schedules, costs, advantages, and disadvantages for the recommended solutions.

Volume III of the final report provides a detailed functional specification for systems comprising the recommended solutions.

SECTION 2

COINS MULTI-LANGUAGE PROBLEM DEFINITION

Currently, a COINS user must employ three different user data languages to query the 48 intelligence data files provided through COINS I. COINS II is scheduled to be expanded by one new system per year so that the intelligence community will be able to access 100 – 150 files supported by 8 – 10 systems within the next five years. Unfortunately, each of these systems has its own user data language, with the result that COINS users will be required to learn 8 – 10 user data languages if they wish to query these files. Further, because each user data language is implemented using a different data base management system (DBMS), the interpretation for a query even to duplicate files on different systems will vary. Finally, because each system supports different types of terminals, the user is forced to learn terminal unique characteristics in order to access different systems within the community.

This problem can be seen even more clearly by examining some of the characteristics of the user data languages and DBMS which COINS users currently or potentially must use in order to access the COINS intelligence data files.

2.1 COINS USER DATA LANGUAGES

A user data language consists of commands which allow the user to direct a specific target system to perform the functions of file creation, data definition, interrogation, manipulation, updating, and display. The user data languages currently available through COINS (ISS (DIAOLS), PIRL, and TILE) only provide the functions of data interrogation, manipulation, display, and limited

updating to their users. User data languages which are potential candidates for addition to COINS include CIRC-II (Air Force Foreign Technology Division), DSRS (State Department), GIM-II (CIA system), M-204 (NSA), DMS-1100 (NPIC PIRL replacement), and SOLIS (NSA). Of these data languages, only DMS-1100, GIM-II, and M-204 provide capabilities beyond interrogation, manipulation, updating, and display — and these additional capabilities require the use of special purpose functions and languages which are separate from the normal user data language. This lack of complete and uniform capabilities in the user data languages currently or potentially available through COINS is one aspect of the COINS multi-language problem.

An even more serious aspect of the problem is the existence of multiple command names to designate identical functions. This part of the COINS multi-language problem is illustrated in Figure 2-1.

Finally, each current and potential user data language has its own way of constructing record selection expressions and punctuating commands. These aspects of the multi-language problem are shown in Figures 2-2 and 2-3.

2.2 COINS DBMS

A data base is defined to be a collection of one or more files containing the occurrences of multiple record types comprising the relationships between records, data aggregates, and data fields. A DBMS is a collection of software and/or hardware with the capabilities needed to define, create, query, update, manipulate, and display all or some portion of a data base. Each of the DBMSs currently or potentially included in COINS are distinct from one another in spite of the fact that they provide the same basic functions; e.g., interrogation, update, display, etc. The differences between the current and potential COINS DBMS lie in three areas: logical data structures supported, physical data structures supported, and the operational characteristics of the functions provided by each DBMS in response to user commands.

User Data Language	Interrogation	Manipulation	Updating	Display
CIRC-II	SEARCH, SELECT	SORT, SAVE, RANK, PURGE, CHANGE	N/A	BROWSE, MAIL, PRINT
DIAOLS	REQUEST, REFINE, TEXTSCAN, RELATE, GEO, CIRCLE, ROUTE	COMPOSE, DSKOUT	UPDATE, CHANGE, ADD, DELETE, MASS	OUTPUT, REPORT, DISPLAY, COMPOSE
DMS-1100[1]	FIND, COUNT	DEFINE, CALL, SAVE, PURGE, INVOKE, ROLLBACK, HOLD, RELEASE	CHANGE, DELETE, INSERT, REMOVE CREATE	LIST, FORMAT, OUTPUT
DSRS	SEARCH, LIST	N/A	N/A	VIEW TEXT, PRINT, TEXT, VIEW CITATION, DISPLAY
GIM-II	FOR	N/A[2]	ADD, DELETE, CHANGE, DELETE-FILE	LIST
M-204	FIND	FOR, IF, IN-CLUDE, JUMP, PLACE, RE-MOVE, SORT, COUNT, CLEAR FILE RECORDS	STORE, ADD, DELETE, CHANGE	PRINT, RESET, SET SKIP
PIRL	INTER, ALSO, COUNT, GET	N/A	N/A	PRINT, DISPLAY
SOLIS	"ADD-screen," "OR-screen," "free-screen"	N/A	N/A	"send-key"
TILE	EXT, EXTI, EXTG, EXTIG	SORT, USE, CONST, INVOK, COMPUTE[3]	UPD/ADD, UPD/DEL, UPD/SUB UPD/SAVE UPD/DELCR	PRINT, OPTION, FORMAT, BUILD, PUBLISH

[1] Query language functions.

[2] A noninteractive procedural-oriented-language (POL) is available.

[3] Subordinate to BUILD command.

Figure 2-1. Commands by Function is User Data Languages

User Data Language	General form of Relation Term	Relational Operators (RO)	Logical Operators (LO)
CIRC-II	value LO value; field-name RO value	EQ, GT, LT, WL, OL	AND, OR, NOT, XOR, PAR, SEN, ADJ
DIAOLS	field-name RO: value:	EQ, GT, LT, BTW, CONTAINS HAS	AND, OR, NOT
DMS-11	field-name RO 'value'	EQ, GT, LT, GE, LE, WRG, ORG	AND, OR, NOT
DSRS	value (field-name)	=[1], THRU	AND, OR, AND NOT
GIM-II	field-name RO "value"	EQ, TG, LT, GE, LE	AND, OR, NOT
M-204	field-name RO value	=, IS, IS GREATER THAN, IS LESS THAN, IS BETWEEN	AN, OR, NOR, NOT
PIRL	index term, fieldname index term b[2] value field name b[2] value	implied =	,
SOLIS	field-name value	TO, = *	AND, OR, WITHOUT
TILE	field-name RO (value); field-name (value)	=[1] >, <, TO	AND, OR, NOT, ALSO, UNLS, AND EITHER

[1]Implicit usage.

[2]b = blank

Figure 2-2. Record Selection Expressions for User Data Language

User Data Language	Typical Command Formation and Punctuation
CIRC-II	.. SELECT 02 01 SEQNBR > 100 AND DOCVOL = 25
DIAOLS	REQUEST NEW NAME EQ: JONES: OR SALARY GT:20000:
DMS-11	COUNT NAME SALARY WHERE NAME EQ'JONES' OR SALARY GT '20000''
DSRS	SEARCH NIXON (SP) OR KISSINGER (SP)
GIM-II	FOR EMP WITH NAME EQ "JONES" OR SALARY GT GT "20000" LIST JUST #
M-204	1. FIND ALL RECORDS FOR WHICH NAME = JONES OR SALARY IS GREATER THAN 20000
PIRL	INTER, IDF; ICOUN, XYZ.
SOLIS	WINDEX GAS WITHOUT TILDEX DISASTER
TILE	EXTI/EMP; NAME (JONES) OR SALARY > (20000)

Figure 2-3. Command Punctuation in User Data Languages

The differences in logical data structures supported by each current and potential COINS DBMS are illustred in Figure 2-4.

The physical storage structures supported by each DBMS are a function of the host system hardware, operating system, and local node conventions.

The differences in operation of a function such as interrogation vary from system to system. For example, in one system interrogation will bring selected records into a work area for further examination while in another system only a list of the addresses of the selected records are retained.

2.3 RESTATEMENT OF THE COINS MULTI-LANGUAGE PROBLEM

Given the foregoing set of facts, the COINS multi-language problem can be restated as follows:

> "What is the best way to provide to current and future COINS users in the intelligence community a single data language which has standard command forms and functions over all the nodes served and accessed through COINS?"

COINS DBMS Systems	Standard Data Entities								
	Fields					Aggregates		Record	File
	Single-Valued	Multi-Valued	Concatenated	Name Multi-Valued	Indexed	Array	Repeating Group		
TILE	FIELD	–	–	FAMILY			(FORMAT)	RECORD	FILE
M-204	–	MULTIPLY OCCURRING FIELD	–	–	–	–	–	RECORD	FILE
DIAOLS	ELEMENT	PERIODIC ELEMENT	SYNONYM	–	RELA-TIONAL PERIODIC ELEMENT	–	–	RECORD	FILE
GIM-II	SINGLE-VALUED FIELD	MULTIPLE-VALUED FIELD	CONCATEN-ATED	–	–	–	D1/D2 RELATION-SHIP	ITEM	DATA LIST
DMS-II	DATA-ITEM	–	DATA AGGREGATE	–	–	TABLE ARRAY	(RECORD TYPE)	RECORD	AREA
DSRS	(CATEGORY)	(CATEGORY)	–	–	–	–	–	CITATION DOCUMENT	FILE
SOLIS	(RETRIEVAL STRATEGY)	(RETRIEVAL STRATEGY)	–	–	–	–	–	MESSAGE	VOLUME
CIRC-II	FORMATTED FIELDS, OUTPUT ELEMENTS	–	–	–	–	–	–	DOCUMENT	DATA BASE

Figure 2-4. Logical Data Structures Supported by Current and Poential COINS DBMS

SECTION 3

ANALYSIS OF COINS REQUIREMENTS

The requirements to be satisfied by any solution selected for the COINS multi-language problem were determined by analyzing:

a. the types of users supported by COINS,

b. the user data language needed by these users,

c. the DBMS needed to provide the data structures, functions, and performance as required by the defined user data language, and, finally,

d. the user-language-DBMS-network interfaces.

3.1 USER REQUIREMENTS

User requirements which must be satisfied by the solutions to the COINS multi-language problem were derived by identifying representative COINS users and determining the functional and operational requirements for each type of user. These functional and operational requirements were determined by analyzing previous studies both internal and external to the COINS project; interviews with COINS users, training specialists, operations personnel, and a variety of data management system specialists in and outside the intelligence community; and through meetings with the COINS multi-language study advisory committee. An interview bibliography is shown in Appendix A of this volume.

3.1.1 User Types

Five types of users can be identified for COINS: casual-user analyst, active-user analyst, retrieval technician, file sponsor, and data management specialist.

The casual-user analyst is an intelligence analyst whose formal training and work experience is in some area other than computers. The casual-user analyst accesses COINS directly or indirectly. The casual-user analyst who accesses the COINS data bases indirectly uses a data specialist to actually query COINS. The casual-user analyst who accesses the data bases provided by COINS directly does so through an online terminal (CRT, TTY, keyboard entry, or RBT). The casual-user analyst knows about one or more data bases maintained by COINS, but in general does not know the detailed contents or structure of the data in the files which make up the data base (the NPIC files represent a data base known to casual-user analysts). The casual-user analyst who is willing to access COINS directly in general only knows one user data language. The casual-user analyst learns and uses the user data language for the system which manages the data bases he uses in his work. His use of a user data language is restricted to simple query and display commands. In addition, this user may create working files for his own use, but does not define data structures for these files, data manipulation procedures, complex queries, or data display procedures. If the casual-user analyst wants to develop complex queries, data manipulation, or data display procedures, he asks a retrieval technician or a data management specialist to prepare these procedures for him, and they are provided to him either as "new" commands (e.g., USE in TILE) or as punched paper tape which he can enter at his terminal.

The active-user analyst is an intelligence analyst whose formal education and work experience is in some field other than computing. The active-user

analyst accesses the data bases provided by COINS directly through an on-line terminal. The active-user analyst knows and employs one or more user data languages and several data bases. The active-user analyst tends to stay in one user data language for most of his work, but will employ other user data languages as needed. The active-user analyst's use of his primary user data language is concerned with interrogation and display, but will also include the activities of working file creation; data definition for these files; parameterized query, simple data manipulation, and display process development; and displaying data in online and off-line modes. The active-user analyst's use of other user data languages is generally restricted to simple query and display requests. The active-user analyst, while not a data management specialist, is thoroughly at home with a terminal and expects both good performance and quality dialogue from the system that he is accessing. The active-user analyst participates actively with the data management specialist to develop "new" commands and data base capabilities.

The retrieval technician class of users are those individuals who either through formal training or work experience have become expert in the use of one or more user data languages supported by COINS. Examples of this type of user are computer specialists or interns and administrative personnel such as data clerks. These users work, in general, on behalf of other users. The retrieval technician accesses files directly through online terminals. The retrieval technician knows more than one user data language, and is expert at forming queries and display commands in these languages. The retrieval technician may or may not have working knowledge of the files that he is accessing. As a result, while the retrieval technician's queries and display commands are well formed, they may not produce the data

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

desired by the retrieval technician's client due to the retrieval technician's lack of understanding of the application.

A file sponsor is responsible for the definition, creation, maintenance, and management of one or more files on a system in COINS. A file sponsor accesses the system containing his file(s) with online or remote batch terminals and through the operations staff of his host system. The file sponsor knows only the user data language(s) supported by the system containing his files. He or a data management specialist working in his behalf use the user data language(s) to define a file and its structure, create the initial version of the file via batched data entry, update the file contents with both batch and incremental transactions, and manage the file by establishing user access rights, update schedules, and file content validation after update.

The data management specialist is responsible for the development, operation, and maintenance of specific data base systems accessible through COINS. The data management specialist accesses his data base system(s) through online and remote batch terminals and through the operators of the computer system which supports his data base system. The data management specialist works at two levels: definitional and operational. The definitional work involves building commands, files, and data structures for use by other users of the data base system. The operational work consists of changing job priorities, aborting jobs, invoking backup storage of files and data bases, establishing and maintaining data bases, instituting recovery procedures because of system or user error, and communicating with users of the system. The user data language used by the data management specialist allows him to declare named files, define data structures over these files, define data manipulation processes, accomplish online or batched data entry, establish and activate queries, display data randomly or in report form,

establish access and privacy controls, and monitor and maintain his data base system.

3.1.2 User Functional Requirements

User functional requirements were developed for each type of COINS user in terms of the following functional capabilities: system control, data definition, file creation, data interrogation; data manipulation, file updating, data display, and user support. In addition, levels for these requirements were established as primitive, standard, and extended. Each of these functional requirements is described in the remainder of this section, and a chart summarizing these functional requirements by type of user and level capability is shown in Figure 3-1.

FUNCTIONAL CAPABILITY	TYPES OF USERS				
	CASUAL USER	ACTIVE USER	RETRIEVAL TECHNICIAN	FILE SPONSOR	DATA MGMT SPECIALIST
1 SYSTEM CONTROL	P	S	S	S	E
2 DATA DEFINITION		P	P	S	E
3 FILE CREATION		P	P	E	E
4 DATA INTERROGATION	P	S	S	S	S
5 DATA MANIPULATION		P	S	P	E
6 FILE UPDATING		P	S	E	S
7 DATA DISPLAY	P	S	S	S	S
8 USER SUPPORT	E	S	P	S	P

LEVEL OF CAPABILITIES	SYMBOL
PRIMITIVE	P
STANDARD	S
EXTENDED	E

Figure 3-1. Functional Capability by User Type

3.1.2.1 System Control. All COINS user types need the functions for "log-on," "log-off," indicating user identification and passwords, designating which node or data base and file is to be accessed, etc.

3.1.2.2 Data Definition. Certain COINS user types need data base definition facilities. These user types are the active-user analyst, the data management specialist, and the file sponsor.

The active-user analyst needs the functions required to create simple working files. These functions include requesting a file, assigning a name to the file, and defining simple schemas for the file. Defining a simple schema consists of identifying the field and aggregates which comprise the file, declaring the types of values to be associated with each field or aggregate, defining how the fields and aggregates are to be composed to form record types, and, finally, assigning the security conditions needed for the file if it is to be shared by other users.

The data management specialist requires functions which allow him to:

a. "Name" hierarchical and linked files.

b. Allocate storage for these files in a host system.

c. Select the storage structures to be applied to the file.

d. Identify the fields, aggregates, and records which comprise the file.

e. Assign attribute and value types to each element.

f. Define how the fields are to be composed to form aggregates, the aggregates to form other aggregates, the aggregates to form records, and the composition of relationships both static and dynamic over groups and records.

g. Establish security privileges for the file.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

h. Define rollback and recovery procedures.

i. Establish file usage monitoring for the file.

The file sponsor has requirements similar to those for the data management specialist.

3.1.2.3 File Creation. Once a data definition has been developed, actual instances of the file(s) which comprise the data base can be created using this data definition. The functions required to accomplish data base creation include:

a. Data definition of the input source file(s).

b. Allocation of media space for the resulting data base files.

c. Translation and acceptance of input data files generated on other computers or under different operating systems.

d. Translation and acceptance of input data files generated on host system but not under the COINS data base manager.

e. Acceptance of files constructed by use of interrogation functions on existing network files.

f. Acceptance of data input as a stream of transactions to intermediate file and then transferred to COINS data base system.

g. Control and acceptance of multifile input.

The COINS user types primarily requiring these functions are the data management specialist and file sponsor, although the active-user analyst may also need them.

In addition to the forgoing file creation capabilities, COINS users require a number of data entry functions. The data entry functions required by casual-user analysts and administrative users consist of system-directed data entry with and without data validation. Directed data entry allows the user to enter

data via a keyboard entry device, with the system displaying the name of the data to be entered and indicating to the user where to enter the data with a flashing cursor or printed character; i. e., a question mark following the name of the data quantity to be entered. Excellent examples of directed data entry are demonstrated by the DSRS and SOLIS query modes.

Currency and accuracy are two hallmarks of a good data base. One way to ensure the continuing accuracy of a data base is through data entry validation. Data entry validation consists of comparing data upon entry with one or more pairs of range values. If the data being entered do not fall within specified validation ranges, the user is notified for confirmation of the data, a change-allowable data range, or correction of spurious data. In addition, data editing facilities which allow numeric and alphabetic tests on data at entry are required.

Both directed and nondirected data entry facilities with and without data editing and validation are required by the active-user analyst, data management specialist, file sponsors, and certain operations personnel.

3.1.2.4 Data Interrogation. Data interrogation requirements by COINS users consist of two primary functions: selection and extraction. Selection is concerned with the location of a specified data element at a specified level of logical organization. This ability is the direct result of the degree of precision attained by the conditional expression in establishing search criteria. A conditional expression, for instance, may be composed in such a way that any field from any hierarchical level in a file may be located, or so that once a field in a particular data aggregate has been found all data aggregates subordinate to that data aggregate qualified by the search field can also be located for further processing.

Data extract is often not clearly delineated as a required function because selection and extract are not mutually exclusive. Often extract is regarded as essentially the copying of selected data before they are formatted into a readable report. In other cases, the term "extract" is treated synonomously with "select," and both have the same set of attributes. However, the two concepts can be more clearly separated for the purpose of this analysis by postulating two different sets of attributes. Selection may be said to involve the depth of the search achieved within the limitations imposed by the rules and language tools available for composing a conditional expression. Extraction involves considering which quantities at each level (field, data aggregate, record, and file) can be searched for and placed in the report or output file. Extraction also decides how many files or reports associated with the same file can be output for the same conditional expression. The allowance for multiple outputs may also involve deciding the directing of data to different output media.

Additional capabilities required by COINS users in the area of data interrogation enable the user to produce or create data values and retrieval specifications at one point in a sequence of operations to be used at a later time; e.g., during output presentation. This capability includes the production of temporary files which can be used as an input file in another statement or to produce additional copies of previous outputs or reports; the creation of prestored retrieval specification; the modification of those specifications; and the parameterized execution of those specifications.

3.1.2.5 Data Manipulation. The primary data manipulation capabilities required by COINS users are the functions necessary to define and create processes for manipulating data and invoking previously defined processes. In the case of the active-user analyst, these functions can be supplied through

a set of computational commands such as provided by a simple form of the BASIC language. The data management specialist requires an extended form of the BASIC language which admits the notions of sub-schemas. Once a data manipulation process has been defined, functions are required which allow it to be added to the users catalog, listed, edited, called for execution, etc. In addition, COINS users each have requirements for standard data manipulation functions such as sorts, statistical and time series analyses, etc.

3.1.2.6 File Updating. Updating a file is the process of using update data to change values in all records or selected records, data aggregates, or items stored in the file. It does not include changing the logical data structure, data validation criteria, or security procedures (alterations of this type are usually made by revising or redefining the data definition of the data base). The file update functions required by COINS data base systems consist of data definition functions which supply descriptions of the section of the data in the data base to be updated and by file creation functions which allow data currently stored in the section of the data base to be updated.

COINS data base systems must provide several modes of file update to meet the needs of the different user types. These modes will differ in terms of user control, input media used, language form used, and functions provided. For example, one user, depending upon his file access rights, needs a file update mode where update processing can be specified for any of the data in the file and another user needs a mode where changes can only be made to certain items or fields in a record.

File update functions consist of three elements: the update data (transactions), the description of the update data to be applied to the data file (transaction definition), and the processes that apply the update data to the data file.

These elements have several sets of attributes which can be varied to meet the file updating requirements of COINS users. One set of attributes common to all three is their source or the way in which users enter them into a data base system. The input medium available for update transactions must be considered as well as the time at which the transaction definitions and update processes must or can be entered.

File sponsors using batch processing to update their files will require functions which allow transaction definitions and processes to be supplied at the beginning of the batch or at least permit them to be included with the transaction. Another attribute closely associated with sources of input and modes of processing is the ordering of transactions. COINS users require functions which allow both random and ordered updating based upon a defined ordering criteria.

Additional functions required for file updating include transaction definition and transaction processing definition. Attributes of transaction definition include format and placement. The transaction may be defined in one of several available formats, such as narrative or fixed tabular, which can also be employed for the transaction program applied to the same transaction. The format(s) in some systems are prescribed, while in others any format may be acceptable as long as the definition has been previously stored and is accessible by the system. The placement of the definition is another variable. A language capability may be provided that permits transactions to be submitted to the system in a self-describing form. In this case, the transaction and its definition are intermingled. On the other hand, especially in systems which use the same medium for both the transaction and its definition, a complete separation of the two is required, the definition preceding the transaction. Such techniques may vary within the system according to

mode or according to the particular transaction data element (field, data aggregate, or record) that is named.

Format and placement are also attributes of the transaction process. In some systems, the transaction process may be intermingled with the transaction definition in the same format. In others, the definition must have been previously entered into the system. Data mapping, whereby the transaction aggregate and field names are equated to corresponding master file aggregate and field names to which they correspond, is another possible attribute of the transaction process. In some systems, this is achieved by using file aggregate and field names in the transaction definition for corresponding data elements rather than using transaction processing statements. Use of the transaction definition for this purpose is usually not possible, however, when data files created for other purposes are used as transactions.

Finally, transaction validation, editing, and transformation is also required and can be accomplished either through the transaction process or through the transaction definition facility. Validation may include simple checks on field values to ensure that they are within the limits supplied by the user in the transaction definition. Conditional expressions used for the interrogation function may also be available for the user to specify, for example, logical relations that must exist within a transaction before it is applied to a file, or logical relations that must exist between the transaction and the file data before the transaction can be applied to the file. Editing is usually provided through the data definition and file creation facilities. Transformation may also be supplied through encoding and decoding subtables specified at transaction definition time or through computational algorithms applied to the data during the transaction program.

3.1.2.7 Data Display. Data display is the formatting of the results of the user queries to a data base system into readable reports or into machine readable form for further processing. This paragraph describes two modes of data display required by COINS users and enumerates the report formatting capabilities of each which must be provided by COINS data base systems.

The two modes which are required by COINS users are the standard system-supplied formats which are an integral part of the system and should be provided to the user either automatically or upon request, and the user-composed report mode (also called the report-generation mode) which provides a means for the user to compose his own output formats and should be tailored to provide the exact output formats and contents desired. The report generation mode must, for instance, allow the user to cause the output to conform to preprinted or to extremely wide output forms. Both modes can be available under recurring scheduling (production at a specified frequency) or demand scheduling. Validation and editing capabilities should also be available to both modes.

a. Standard mode. Specific attributes of the standard output mode include:

1. A standard set of report types: i. e., the way all data are presented (tabular, row).

2. Automatic calculation and adjustment of the format parameter to the output device; e. g., the presentation of output in balanced columns which are automatically adjusted to correspond to the maximum number of characters per line that can be printed by the output device.

3. Parameter specification; e. g., the system can select and print headings provided by the user, such as report title, column headings, data, time, and page numbers.

4. Special functions specified by the user, such as sums and counts of retrieved item names and values if specified by the

user; counts and tallies of given item occurrences; statistical operations on given values; and editing and decoding operations.

b. Report generation mode. The report-generation or user-composed report mode relies on specific report creation capabilities provided by the data base system to the user for the data to be output. These capabilities can include:

1. Use of literal values and field values for headers.

2. Pagination control, whereby the data base system provides for user-specified parameters for controlling and numbering individual pages of an output report (examples include starting page specification wherein the user is allowed to specify the starting page number and the increment to be used in determining the page number of succeeding pages, and page break whereby the user is allowed to specify certain methods to stop printout on a given page and begin a new page independent of specified line counts).

3. Data reduction features, whereby a data base system allows user-specified sums, counts, and statistical operations to be performed on specific item names or values.

4. Use of page headers and trailers.

5. Output, whereby a data base system allows the user to request multiple copies of a report or to request output to conform to certain output forms.

6. Special outputs, whereby a data base system provides to the user, upon request, special output such as job summaries on a high-speed printer or at a terminal.

3.1.2.8 User Support. A number of user support functions are required for users of COINS data base systems. These functions can be divided into two broad categories: direct and indirect user support. Direct user support functions include "command help" and "data help" capabilities. Indirect user support functions include system control capabilities consisting of monitoring, error recording, restart and recovery procedures, and security.

The command help function allows a user to enter the word "help" followed by the name of the command with which he would like assistance.

The data help function allows a user, based upon his access rights, to be shown a human-oriented data definition for the file he is accessing. This humanized data definition would display for the user the names and relationships of data that he is allowed to access. In addition, this function should provide thesaurus capabilities so that the user can enter his reference names and receive probable equivalent data base reference names.

The monitoring function provides an optional recording of data base system activity. Recording control pertains to the use of recording categories which define a particular collection of events to be recorded. Every event to be monitored in the data base system is assigned to a category. The highest system-recording category records all events defined. Lower categories record a particular subset of specified events. A system usually provides for varying the recording frequency based on the system recording category in operation or upon such criteria as specific time of day and specific sampling period.

Information gathered covers data base access and program module use, and can cover general statistics and specific facts. General statistical

information provided by monitoring includes tallies of events, such as total items retrieved, and tallies of the number of consoles in use and the number of disc seeks issued; total times for events, such as total time for processing a specific job or total time required to search the file; and standard job accounting or job history, which may supplement standard operating system accounting.

Specific information can be provided by two modes of monitoring: demand and background. Demand monitoring involves the surveillance of a condition or conditions specified by the system manager and provides for both standard and specialized queries of system activity at the discretion of the system manager.

Standard queries report on what devices and modules are currently in use, what type of file is being accessed, what users are currently active, and what amount of working storage is available. Specific queries single out particulars of system activity such as data accessed by a specific user within a given time period or the time of the last update performed on a specific file.

Background monitoring occurs on a continuing basis to maintain system regulations. It informs operations personnel of the existence of abnormal system use, such as attempts to access the data base illegally, or the reason for abnormal termination of runs.

Error-recording functions should also be provided for data base accountability, the assurance of processing integrity, the full identification of errors, and the correction of error conditions. All user-initiated communications, including data input, are subject to system error detection and validation such as checking for format, syntax, and semantics. A data base system should be able to record errors by the operating system, such as

hardware-generated errors. The provision for diagnostics assists the system user in checking out new procedures and developmental files. This provision is actually a test mode of operation which facilitates debugging operations and protects a system against possible simple and catastrophic failures resulting from test failures.

The restart and recovery functions should provide the capabilities necessary for recovery from interrupts and programmer, operator, and hardware errors. To provide for the event of catastrophic failure, the system should also be able to maintain a backup data file.

Security functions are absolutely mandatory in COINS data base systems to protect the information contained in their data bases from illegal access. These features should include restriction of access through invoking read/write protection at various levels of data, automatic destruction of data on any storage device in the event of imminent compromise, automatic clearing of any area of core containing classified information immediately after the last use of that information, and assignment at file definition time of access locks or security codes to files, records, data aggregates, and fields. Of further interest to the user is the time security clearance takes place, the levels of data at which security restrictions can be defined by the user, the extent of security restriction application (data access and/or data modification), and the security clearance procedure to be satisfied by the user himself before he can execute any data handling statements.

3.2 LANGUAGE REQUIREMENTS

The primary language requirement for COINS users is that each user should be able to access the data base systems of his choice using a single user data language. Further, the user data language for a given user type should remain invariant with respect to device; i.e., online terminal, remote batch terminal, or local batch submission. Finally, the user data language

supported by a data base system in COINS should be the same for all users at a node and on the network (ISS in DIAOLS vs. the DIAOLS COINS user language).

Specific user data language requirements were determined from the standpoint of the structure, dimension, extension, transformation, and representation capabilities needed for a COINS user data language.

The structure of a data language should consist of short declarative, imperative, and interrogative phrases in order to meet the requirement for a simple, easily learned set of commands and allow online translation of these commands. These phrases should consist of verbs (commands) followed by argument lists.

The dimension requirement is derived from the different types of users for COINS. A data language should provide at least four levels or dimensions of capabilities as follows:

a. Simple commands with simple arguments.

b. Simple commands with complex arguments.

c. Complex commands with simple arguments.

d. Complex commands with complex arguments.

By defining this kind of dimensionality for the language, arranging for this dimensionality to be represented syntactically, and assigning capabilities to each user based upon the dimensionality, each user's access to the system can be controlled and supported more effectively.

The extension requirement means that a data language should contain the capabilities for its own extension by certain of its users.

The fundamental transformation requirement placed upon a user data language for COINS is that it be able to represent any statement from a selected target query language as a sequence of user data language statements.

The representation requirement is similar in that a user data language for COINS must provide the capabilities needed to represent the data structures used by the data base systems in COINS.

Four classes of data structures are required by COINS users. These classes are fields, data aggregates, records, and files. Descriptions of these structures follow.

A field is a single elementary data entity containing no logical substructures. The principal attribute of the field is the value. Other attributes might include identification, type, security, and value existence indicators.

A data aggregate is a set of fields and possibly other data aggregates. Data aggregates can be simple or compound, repeating or nonrepeating. A data aggregate composed solely of fields is called a simple aggregate; an aggregate composed of a set of fields and a set of related aggregates is called a compound aggregate. An aggregate is the lowest level of data structure concerned in the logical organization of a data base; it can maintain at once three different relationships with other aggregates: parent (superior), dependent (subordinate), and peer. Within a compound aggregate, aggregates can be manipulated to establish a hierarchic organization of these relationships. (The term "hierarchic" implies that each parent aggregate instance can be optionally paired with one or more dependent aggregate instances.) The kinds of hierarchical relationships that can be organized by the user depends upon attributes of aggregate composition; i.e., the way single aggregates are arranged to compose another structure.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

An aggregate relation is the logical relation or mapping between two sets of aggregates, the first set being the parent aggregates and the second set being the dependent aggregates. An aggregate relation has a set of attributes of its own. The relationship facility provided is equivalent to the hierarchic aggregate relationship facility provided in a compound aggregate, except for the following differences.

In a compound aggregate, an aggregate may be subordinated to one parent aggregate only; in an aggregate relation, an aggregate may be subordinated to many parent aggregates. Such a relationship is often termed nonhierarchic.

In a compound aggregate, the principal fields (the immediate constituents of a compound aggregate to which an aggregate is subordinated) do not have a collective name. In an aggregate relation, each set of fields to which an aggregate is subordinated may have a name of its own; namely, the parent aggregate name.

An aggregate relation may consist of a single repeating or nonrepeating parent aggregate entity and a single repeating or nonrepeating dependent aggregate entity. On the other hand, an aggregate relation may have multiple dependent aggregates. In this case, the instances of the different dependent aggregates can be treated as members of a single set whereby they may be processed jointly in one context as a single set, or they can be treated as members of several different sets whereby they may be processed independently in another context. This characteristic is not present among compound aggregate attributes.

If the aggregate relation is provided in a given system, it may be an explicitly defined structure type or it may be defined as part of a larger structure type. In either case, the structures possible in the system for aggregate

relations are composed, or the way they are used to compose other structures can vary.

A record is a particular set of aggregates in which one, and only one, designated aggregate is not contained in or subordinate to any other aggregate. The record is the primary unit of data activity in a COINS data base system.

A file is a set of records that have the same logical organization. The file corresponds to a set of application entities, such as countries, governments, projects, or organizations. The entities of a file may be from the same class; e.g., projects and organizations. Its records may be explicitly interrelated, or related only by ordering and otherwise independent of each other. The aggregation of all files which can be accessed by a DBMS is a data base.

3.3 DBMS REQUIREMENTS

Requirements which must be satisfied by any DBMS adopted as a standard for COINS have been developed. These requirements are divided into five major categories: data structures, processing, input/output, operational support, and environmental.

3.3.1 Data Structures

A standard COINS DBMS must support data structures with the following general characteristics:

a. Data entities equivalent in capability to field, data aggregates, records, and file must be provided.

b. File organizations equivalent to sequential, indexed sequential, hierarchical, and inverted list structures are needed for the COINS files. In addition, network structures are also needed to effectively handle graphic processing.

c. Data formats are to include fixed and variable length fields and records for both formatted and text data.

3.3.2 Processing

The processing capabilities required for a COINS standard DBMS are as follows:

a. Processing modes — must support three modes of processing — interactive, remote batch, and local batch — for both random and sequential files.

b. File definition and creation capabilities — which allow new files to be defined and created in response to user commands.

c. Data entry and editing — which allow input data to be entered by the user in all modes of processing and edited by the system including testing range of values, specifying valid code conditions, and performing numeric or alphabetic tests on the data.

d. File updating — which allows certain users to add, delete, or insert new fields or records within a file or change data contained within these fields or records.

e. Data interrogation — capabilities which support simple to complex queries including multi-file searches and multi-file cross-referencing.

f. Audit trail and transaction statistics — facilities must be available to automatically produce a file of all transactions which affect the data stored in designated files including the capabilities to provide statistics on the number of records added, inserted, changed, or deleted from these files.

g. Arithmetic capabilities — must include addition, subtraction, multiplication, and division.

h. Logical capabilities — complete relational and boolean operations.

i. Demand reporting facilities — which allow unanticipated reports to be rapidly designed and obtained.

j. Multiple file cross-referencing — which allows the system to use fields of data in one file to locate records in other associated files for subsequent processing.

k. Interface to other high level languages — the ability to process files which are created in the system using other high level languages; i. e., the DBMS should provide the same interface to all other systems supported on the computer.

l. Independent of file changes — should have capabilities for remaining independent of changes in the structure, content, or format of data files; e. g., the schema/sub-schema approach of CODASYL DBTG.

m. Data base expandability — capabilities are needed by the COINS DBMS for adding new data files to a data base and establishing or revising linkages between previous files and the new files.

n. Cataloguing of requests — capabilities are needed to store user command sequences in files and activating these previously defined sequences by name only. In addition, run time parameters should be allowed.

o. File restructuring — facilities are needed to allow the COINS DBMS to change the file structure (i. e., sequential to indexed sequential, etc.) and also to change the record structure; e. g., fixed to variable.

p. Back-up file capabilities — required for creating back-up files in case of loss of data through system failure or file destruction.

q. Message switching capabilities — needed for routing messages to and from terminals in response to transaction requests and status reporting.

r. Logging and accounting capabilities by terminal — must be provided by the COINS DBMS or its supporting operating system.

s. Message time stamping capabilities — for indicating date and time on messages which are sent or logged by a file in the system.

3.3.3 Input/Output

The input/output facilities required for a standard COINS DBMS include:

a. Large files — the system must be able to easily and efficiently handle numerous files containing a large number of variable length records; e.g., 574 million bytes with 64 to 10,000 bytes per record.

b. Report capabilities must include:

  1. Editing capabilities for output data fields including leading zero suppression, floating dollar signs, etc.

  2. Headings specification of titles, subtitles, and column headings, with default options for automatic column heading, page numbering, data specification, and document classification.

  3. Formatting — providing a high degree of control for users in determining maximum field sizes permitted and data placement.

  4. Summary lines and control breaks which allow users to define number and level of breaks permitted and the type of information that can be contained in summary lines such as counts, totals, averages, etc.

c. Foreign file handling — the capabilities needed to read and process files created on other systems and the facilities for converting these files to the type of file structures supported by the COINS DBMS.

d. Report file sorting — the facilities needed to develop special reports by sorting currently available files.

e. Error recovery and restart procedures — must be provided by the COINS DBMS so that processing of data can be restarted in the event of system error or failure from some check or restart point without having to resubmit the entire transaction.

f. Security — provisions must be included within the COINS DBMS to ensure that only authorized users are permitted access to the system. In addition, facilities must be provided to limit access to only those records or fields which the user is authorized to query or update.

3.3.4 Operational Support

The operational support requirements to be satisfied by a standard COINS DBMS include:

a. Adequate documentation — complete and detailed documentation must exist for the COINS DBMS covering three classes of users: data management specialists, system programmers, and operators.

b. Comprehensive diagnostics — must be available with the DBMS covering errors that can occur during compilation, loading, and running the system and its associated application programs.

c. Program modification tools — must be available to allow at least version updating with manufacturer-supplied changes.

3.3.5 Environmental

a. A standard COINS DBMS must be able to run on, or at a minimum be developed for, the following hardware: IBM 370/155 or larger, Univac 1100, Honeywell 6000 series, Burroughs 6/7000 series, and possibly the CDC 6/7000 series equipment.

b. OS release independent — the standard COINS DBMS should be as OS release independent as possible. This may mean only that the standard COINS DBMS supports a standard set of OS/DBMS interfaces.

c. Upward compatible — the standard COINS DBMS must be upward compatible with respect to larger computer systems of the same manufacture, and with changes in peripheral devices and other hardware.

3.4 INTERFACE REQUIREMENTS

Interface requirements which must be satisfied by solutions selected for the COINS multi-language problem were derived in the following areas: network compatibility, COINS II schedule, uniform node data management, minimum node impact, performance, terminal compatability, and cost.

3.4.1 Network Compatibility

Any solution selected for the COINS multi-language problem must be compatible with the communications protocols being established for COINS II. These protocols follow those protocols established for the ARPANET. Furthermore, every attempt should be made to ensure that selected solutions take into account interfacing to individual node hardware. This is particularly important due to hardware changes already in progress at NPIC (Univac 494 being replaced with a Univac 1110) and potential hardware changes contemplated for DIA and NSA (TILE).

3.4.2 COINS II Schedule

Any solution to the COINS multi-language problem must not impact the COINS II schedule, which calls for operation of COINS II at four nodes (DIA, NPIC, NSA (TILE), and NSA (SOLIS)) by end of fiscal year 1976. The addition of one new node to the network is planned each year until the end of fiscal year 1980.

3.4.3 Uniform Node Data Management

Solutions for the COINS multi-language problem should not limit, and optimally provide, the facilities needed to support standard data and file descriptions at each node. In addition, operations over distributed data bases such as multifile searches and updating should be supported. Finally, automatic event notification is also desirable.

3.4.4 Minimum Node Impact

A key interface requirement to be met by any solution to the COINS multi-language problem is minimum impact upon each node in the COINS network. This impact will be felt in three major areas. The first is file conversion

if the solution selected for COINS demands a different file or data structure than that already supported by the node. Another area of concern is the amount of reprogramming of application support software that will be required with the selection of a given solution. DIA has estimated that a major hardware or software change at their node will cost between $5 and 10 million. Finally, careful consideration must be given to the possibility of performance degradation for the users of a node in order for that node to support COINS operations.

3.4.5 Performance

Specific performance requirements were not derived during the requirements analysis portion of the COINS multi-language study. Certain basic guidelines are available as shown in Figure 3-2.

3.4.6 Terminal Compatibility

Currently, nodes within the COINS network and systems which could potentially become a part of the COINS network are supporting a wide variety of terminals. These terminals include the TTY 35; TTY 37; BIDS; Sanders;

	End of FY76	End of FY80
Total Number of Users	500	700 → 1000
Average Number of Users/Node	125	125
Response time		
— Interactive	2 sec	1 sec
— Remote batch	2 minutes	1 minute
Files Supported	48	150

Figure 3-2. Performance Guidelines for COINS II

DIDS; IBM 2741, 2260, and 3270; the Superbee; etc. A requirement for any solution to the COINS multi-language problem is that it must enable the selection of a terminal standard(s) for COINS.

3.4.7 Cost

No precise cost requirements were established for this study. However, the following general observations are in order. First, any solution to the COINS multi-language problem is likely to be expensive. Second, every attempt should be made to align the solution of the COINS multi-language problem with hardware and software changes already in progress or planned at each of the current and potential COINS nodes. Finally, shouldering of the major share of the cost for a solution to the COINS multi-language problem is not likely to be acceptable or borne by the individual nodes.

SECTION 4

ANALYSIS OF CURRENT AND POTENTIAL DBMS

Concurrent with the analysis of COINS requirements, each DBMS currently or potentially a member of COINS was analyzed to determine its characteristics and potential impact on any solution selected for the COINS multilanguage problem. The DBMS analyzed during this portion of the study and their users are shown in the following chart:

DBMS	User
CIRC-II	Air Force Foreign Technology Division
DIAOLS	DIA
DMS-1100	NPIC (replacement for PIRL)
DSRS	State Department
GIM-II	CIA
M-204	NSA
PIRL	NPIC
SOLIS	NSA
SYSTEM 2000	Possible standard DBMS
TILE	NSA

While the primary focus of the analysis was on the user data languages supported by each system, each of these DBMS was analyzed with respect to the following factors: system organization, data structures supported, data definition, file creation, interrogation, manipulation, file updating, data display, security, restructuring, system control, and operational environment. The remainder of this section consists of summary descriptions for each of the foregoing DBMS. Detailed descriptions for each system are provided in Volume II of this report.

4.1 CIRC-II

The Department of Defense (DoD) Scientific and Technical Intelligence Information Support Program (STIISP) is a coordinated, integrated S&T intelligence information system. This system's purpose is to provide intelligence information and support to the following DoD S&T intelligence production agencies:

a. Naval Intelligence Support Center (NISC).

b. Foreign Science and Technology Center (FSTC).

c. Medical Intelligence Information Agency (MIIA).

d. Missile Intelligence Agency (MIA).

e. Foreign Technology Division (FTD).

Although the DoD STIISP is involved with all aspects of information services, its primary service is providing the users with an automated index of publications, reports, and documents containing information of interest to the DoD S&T intelligence production agencies. The most recent version of this automated index is called the Central Information Reference and Control II (CIRC-II) System. CIRC-II provides the user with complete coverage of pertinent documentation, total indexing of information contained in the documentation, and uniformity of processing. CIRC-II provides for the dissemination of new information to a user as well as identification of existing documents. The latter service is provided by the CIRC-II Retrospective Retrieval System.

The CIRC-II Retrospective Retrieval System operates in the online and batch modes. The online CIRC-II user uses an interactive terminal for communicating with the system where, after signing on, he must specify the data base he wishes to interrogate. Any time during this operation, the user may change data bases, or request that certain display outputs be processed off-line.

CIRC-II data structures are somewhat special purpose in nature, being oriented towards document retrieval. CIRC-II is composed of a set of data bases each of which is a hierarchical structure. Although the different data bases are structured the same and have the same logical nodes, their utilization is quite different. The primary data base is the S&T data base which contains all the document representations. The other data bases are called support data bases and are used to facilitate queries to the S&T data base. Each data base is divided into a set of documents, output elements, paragraphs, sentences, and words (in the order of decreasing scope). In addition, special purpose formatted fields have been defined for each data base which facilitate document retrieval.

CIRC-II has two primary interrogation commands: SEARCH and SELECT. The SEARCH command provides basic data isolation at the document level, where word equality searches are made at the output element, paragraph, or sentence level. The SELECT command operates on a set of documents isolated by either a SEARCH command or another SELECT command. This command operates on relational conditions imposed on formatted fields. CIRC-II allows selected display of isolated documents with the BROWSE functions.

4.2 DIAOLS

DIAOLS is an intelligence data handling system developed by the Defense Intelligence Agency. DIAOLS stands for Defense Intelligence Agency On-Line System. DIAOLS is currently operational on two Honeywell G-635 computers supported by the General Comprehensive Operating system. DIAOLS provides for user servicing in three modes: interactive time sharing, remote batch processing, and local batch processing. The primary reason for the development of DIAOLS was to provide a time sharing capability supporting intelligence producer and user needs. DIAOLS consists of five major components: the Intelligence Support System (ISS), Community On-Line Intelligence System (COINS), BASIC, FORTRAN, and CARDIN.

The DIAOLS system resides on dual Honeywell G-635 computers with standard Honeywell 600 peripherals. The terminals available are TTY 37, Raytheon DIDS 400 with Inktronic printers, and G-115 RBT or COPE RBT. The GCOS operating system is used in a modified version to support processing of classifed information. DIAOLS supports three modes of operation: interactive time sharing, remote batch, and local batch. The system may be accessed by TTY-like terminals, CRT displays, and remote batch terminals. Of the two systems, one supports compartmented processing to the top secret/SI level; the other noncompartmented processing.

The kinds of data entities supported by DIAOLS include elements, groups, records, and files. The types of data allowed are standard data, alphanumeric, numeric, and alpha. The logical structuring provided by DIAOLS is hierarchical tree structures using inverted lists. The storage structures used by DIAOLS are indexed sequential and random.

Data interrogation, processing, maintenance, updating, display, and report generation are provided online by DIAOLS. Data definition and data base creation are provided via local and remote batch mode.

4.3 DMS-1100

Data Base Management System 1100 (DMS-1100) is Univac's version of CODASYL's DBTG Data Base Management System. The basic premise of the CODASYL Data Base Task Group is that there must be a separation of data description functions from data access functions. This implies a distinction between the data base manager, whose function is to describe the data records, and the user, whose function is to access the data. This distinction leads to the separate development of a data description language and a data manipulation language.

Currently, DMS-1100 is operational with both the Data Definition Language (DDL) and the Data Manipulation Language (DML). In addition, a new interactive Query Language (QL) is being currently developed by Univac. The

DMS-1100 system resides on the Univac 1100 hardware series where it operates under the EXEC-8 operating system.

DMS-1100 provides great flexibility to the user where simple sequential organization of data to complicated network structures are allowed. These data structures are divided into areas (files), records, and items (fields). In addition, a user may optionally define set relationships over records, where each set is called by name. The record location-mode can be specified by the user where direct, index sequential, calculation by procedure (CALC), and via-set are allowed. A Data Definition Language (DDL) is used by the user to define his data bases.

DMS-1100 provides the user with a Data Manipulation Language (DML) which is embedded in a Univac 1100 COBOL environment. This language allows the user to interrogate his data base (FIND, FETCH, and GET commands), to update or restructure his data base (STORE, MODIFY, DELETE, INSERT, and REMOVE commands), and to conditionally control the sequencing of his DML procedure (IF command). In addition, other miscellaneous data manipulation, data base control, and run-unit control commands are provided, such as OPEN, CLOSE, MOVE, IMPART, and DEPART. The display capability is provided by the standard COBOL display commands. The online interactive Query Language (QL) provides essentially the same capabilities as those described for the Data Manipulation Language.

4.4 DSRS

DSRS is an online document storage and retrieval system in the U.S. Department of State. DSRS stands for Document Storage and Retrieval System. At the present time, the documents stored in the system are the telegrams received electrically from the State Department's ATS system.

DSRS has four parts: 1) document storage, 2) generation and storage of document, indexes, called citations, 3) text retrieval, and 4) citation retrieval

based on selection criteria. DSRS supports two types of terminals. The input terminal is a COMPUTEK CRT. The output terminal is either the CRT or an ASR-37 teletypewriter printer associated with the CRT.

The kinds of logical data entities supported by DSRS include index terms, citation records, and telegrams.

Data interrogation and display are available on line using commands and function keys. The commands are: ACCESS, DISPLAY, ERASE, EXPLAIN, LIST, PRINT CITATION, PRINT TEXT, SAVE, SEARCH, VIEW CITATION, and VIEW TEXT. The function keys are: Display Options, Display Searches, Page Backward, Page Forward, Restore, Send All, and Sign On/Off.

4.5 GIM-II

GIM-II is a self-contained, online, generalized data management system being developed by TRW. The system is currently operational on IBM 360/370 using OS/MFT or OS/MVT operating systems. A previous version of the system, GIM, is in operation on the Univac 11XX series under EXEC-2 and EXEC-8 operating systems. GIM-II supports both interactive and batch processing, but is primarily an online system.

The system is capable of processing several transactions concurrently, and it provides both pseudo-English and procedural access languages. The concurrent transaction processing is facilitated by a resource-sharing executive that manages and allocates all resources assigned to it by the operating system. The pseudo-English access language was retained from GIM, while the procedural language was added to provide FORTRAN, COBOL, and PL/1-like capabilities.

The GIM-II system operates with a collection of different "data bases," some of which are for system operation and others, the majority, are user data bases. GIM-II data bases are hierarchical in structure where they contain the usual file-record-field divisions. Files, which are called by name, are

of two types, those that describe fields (dictionary files), and those that contain data (data files). A master dictionary (also a file) is presented which describes the fields of the user dictionaries.

Fields, which are called by name, can be single or multi-valued, they can contain concatenated values which are referenced as subfields through synonyms, or they can be associated with other fields, either in the same logical record or in separate files. Special preformatted files, called procedure-list files, are used to implement the GIM-II Procedural-Oriented Language (POL), where they are created and maintained as any other user data file.

The GIM-II language can be viewed as a combination of three languages: a pseudo-English data access language, a Procedural language, and a Data Definition language. The data access language consists of components called phrases. There are two basic types of phrases: selection and verb phrases. The selection phrases consist of a selection command, FOR, WITH, or WHERE, followed by a selection criteria specification. The verb phrases consist of a verb followed by a verb specification. The verbs in the data access language include LIST, ADD, DELETE, and CHANGE. The Procedural language consists of a series of commands designed to provide computational and data structuring capabilities to the GIM-II user. The data definition language consists of a series of commands which allow files to be established. Some of the verbs in the data definition language include: STRUCTURE-FILE, STRUCTURE-SEGMENT, DEFINE SEGMENT, and STRUCTURE SYNDICT.

4.6 M-204

M-204 is an online data base management system produced by Computer Corporation of America that operates under OS/MVT on the IBM 360/370 computers. The file storage devices under M-204 include the 2311, 2314, and 3330. Terminals supported by the M-204 system include teletype-compatible devices (hardcopy and CRT), IBM 2741 or Datel, and IBM 2265 or 2260 (remote).

Each system consists of a number of separate segments which may be bound together in various combinations. These segments may be classified into three categories:

a. Routines central to the storage retrieval process which must be present in all configurations.

b. Segments which may be included in a particular job of the user, depending on the purpose of that job.

c. Special purpose segments.

System operation is essentially divided into three levels of control:

a. OS 360/370 "job card" level.

b. CCA Data Management System (CCADMS), the command level.

c. CCADMS, the statement level (the M-204 user query language).

The M-204 system is embedded in a 360/370 system where it resides under the operating system. Once the required data set and series 200 user file definitions are provided through OS 360/370 JCL DD cards, the CCADMS can be initialized. Upon completion of system initialization, the user may create and/or update new files by using the Fast Load (FLOD) program, or he may query current file structures with the user query language. The levels of control between the user query language statements and the system control commands can be switched back and forth with the command/statement pair BEGIN/END. Communication between separate request sequences (delineated by BEGIN/END pairs) is accomplished by a user global variable area which can be manipulated at both the command and statement level.

M-204 data structures are divided into the usual file-record-field components where files and fields are referenced by name. Records can be extracted from a file by specifying selection criteria on the values of fields in the records. Fields can be assigned various types where, depending on the frequency of occurrence of the field and the type assignment, the system's speed as well as storage space can be affected considerably. A record can contain

the same field more than once, which is referred to as a "multiply occurring field."

The M-204 data base manipulation language is composed of primarily two language levels, one called commands and the other user language statements. Commands such as OPEN, CLOSE, BEGIN, SEGMENT, and INCLUDE SEGMENT, to name a few, provide overall control of the user's operation. User language statements are used to manipulate the data base directly. Some of the major statements are as follows: FIND, which provides an interrogation capability; ADD, DELETE, and CHANGE, which give the user an updating capability; PRINT and SET, which provide display and control functions; and IF, JUMP, and PAUSE, which provide overall processing control.

4.7 PIRL

PIRL (Photo Interpreters Retrieval Language system) is implemented upon three Univac 494 computers. The PIRL system supports both interactive and batch operations. Users are able, from TTY 35 terminals, to:

a. Select information from event or installation files and print the information.

b. Count the occurrences of events, installations, or reports pertaining to these events and installations.

c. Compile a list of the events or installations that match a given search criteria.

The query language for PIRL consists of four basic commands: inter, also, count, and get. The inter command counts records which meet all conditions specified by it and prints the total. The also command counts records which meet the last condition and any other single condition specified by the command and then prints a total of the records meeting these conditions. The count command counts all records whose header has a field with a

specified value and prints a total of the records for which this condition is true. The get command not only locates records which meet its query conditions but allows these records to be printed out for the user.

The data structures utilized in PIRL are inverted list structures where each list represents an index to one of the fields or groups within the records contained in the PIRL files.

4.8 SOLIS

SOLIS (SIGINT On-Line Information System) provides online editing, storage, and retrieval of NSA end product documents via a Burroughs B-6700 computer system.

These documents are logically collected by month in "volumes." Currently, 9 months or volumes are online. Three more months will be added until documents covering 12 months will be online. The new documents being added to the current month's volume are never over 30 minutes old. Documents are retrieved using terms, recognizable by SOLIS as index terms, which are kept in one or more system dictionaries. The SOLIS system has been operational since November, 1972.

SOLIS resides on the Burroughs B-6700 computer system. The remote terminals are Burroughs B-9353 CRTs and TTYs for low-speed, hardcopy output. High-speed hardcopy output takes place near the B-6700 on high-speed printers.

Messages enter the SOLIS system and are automatically indexed and made immediately retrievable. Within 24 hours, these same messages are further indexed and added to the current month's volume. Old messages are automatically phased out of SOLIS as new messages come in. All editing and retrieval functions are handled by a software package called the SOLIS/MESSENGER. SOLIS physical data structures are organized by volume, file, record, and word.

The logical data entities are terms from the document and the documents themselves.

Document retrieval and display are accomplished by the user entering the parts of his request on a formatted CRT screen. Basically, key words and/or other document descriptors (such as date of entry into SOLIS) are entered on the CRT and the Send key is depressed. The response from SOLIS is the number of documents satisfying the request. To then view the text of the documents retrieved, the user need only press the Send key again and the first page of the message will appear. Each time the Send key is depressed, the next page of the document will appear.

4.9 SYSTEM 2000

SYSTEM 2000 is a data base management system developed by MRI Systems Corporation to operate on several different computers; namely, IBM 360/370 series, Univac 1100 series, and CDC 6000 and CYBER series. Several operating modes are available to the different types of users (users include the "master password holder" and the general password holder with his various levels of authority) of SYSTEM 2000. These include the Basic SYSTEM 2000, Immediate Access facility, Procedural Language facility, Report Writer facility, and the Sequential File facility.

SYSTEM 2000 consists of the Basic SYSTEM 2000, separate features dependent on the Basic system, a separate system called the Sequential File Feature, and procedure languages based on COBOL, FORTRAN, and assembly language.

Using Basic SYSTEM 2000, the user can issue system control commands having to do with user files, passwords, and requesting either the Define Module or the Access Module. The Define Module allows the user to define the structure of his data in the component/repeating group format. The Access

Module permits, in a batch mode, loading of data into the data base according to the data definition, retrieval of these data, updating the data base, and printing selected data.

Two features which are dependent on the Basic system but not included in it are

a. The Immediate Access Feature, which is similar to the Access Feature (Queued) but is interactive.

b. The Report Writer Feature which allows the user to define one or more reports based on the data.

For Univac installations, the Sequential File Feature provides the user with a way to create and access data bases residing on magnetic tape. The Basic SYSTEM 2000 modules have been modified for this feature and thus the commands often differ in both name and meaning.

The Procedural Language Feature enables the user to manipulate data in a SYSTEM 2000 data base using COBOL, FORTRAN, or assembly language. SYSTEM 2000 procedural language statements are embedded among the COBOL statements and are converted into a series of COBOL statements via a "pre-compiler" provided by MRI.

The data entities in SYSTEM 2000 are repeating groups, user defined functions, strings, records, and data bases. These entities exist in a hierarchical structure of components, data sets, trees, and logical entities.

SYSTEM 2000 allows the user to define his data structure, modify the data structure definition, interrogate and update the data base, perform complex processing of the data, and generate reports. Most of these activities can occur in either an interactive or a batch mode. System security is provided via passwords and levels of authority attached to each password.

4.10 TILE

The Technical Information Processing System (TIPS) was conceived in 1960. By 1965 TIPS was operational on two Univac 490 systems, handling user requests on a 24-hour, seven-day week basis. This was a somewhat restricted system, however, handling only one request at a time and having no standard query language available for the various user data files. Presently, TIPS runs on three Univac 494 computers, where one is completely dedicated for TIPS processing. A standard query language is now available and is called the TIPS Interrogation Language (TILE). Connected to the Univac 494 are 24 Honeywell 516 minicomputers. Each 516 can handle a separate query simultaneously, thus providing TIPS with simultaneity of operation. TIPS operates under the RYE operating system.

The files accessed with TILE are either single-format or multi-format files. The records within both types of files are fixed size. Within the single-format file, all records have the same format, contain the same fields, and are the same size. In a multi-format file, up to 60 formats may be used. Files, formats, and fields are called by name.

The TILE language is composed of statements. TILE statements normally begin with verbs followed by nouns specifying files, fields, and data, related by connectors and relations. The verbs available under TILE include: extract (EXT), PRINT, SORT, CREATE, POST, USE, BUILD, PUBLISH, CONST, INVOK, and UPD. PRINT has its obvious meaning. SORT allows the user to sort the output of an extract in some order other than that which is maintained by the system. CREATE and POST allow the user to create a temporary subfile. USE allows the user to leave the TILE language for special-purpose functions and return to the TILE language. BUILD allows the user to structure canned print format specifications which can be

activated by the PUBLISH verb for subsequent printing. CONST gives the user the ability to construct TILE statement strings which can be activated by the INVOK verb. UPD is the update function for files.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

SECTION 5

ANALYSIS OF ADVANCED TECHNOLOGIES

A key phase in the development of potential solutions to the COINS multi-language problem involved identification and analysis of advanced technologies with probable impact on solution of this problem. Based upon our analysis of solution requirements and the current and potential COINS DBMS, we found that two areas of advanced technology showed the greatest promise with respect to providing solutions to the COINS multi-language problem. These areas are standard user data languages and data base management systems. In the area of standard user data languages, three advanced development projects were identified and analyzed:

a. The SDC Data Sharing on Computer Networks Project.

b. The MIT project on Coupling of Interactive Information Systems.

c. The Uniform Data Language Project at Logicon.

In the area of standard data base management system technologies, four projects were identified and limited analysis performed on them:

a. The CCA Datacomputer Project.

b. The Bell Telephone Laboratories XDMS Project.

c. The IBM 3850 Project

The results of our investigation and analysis of these projects is described in the remainder of this section.

5.1 THE SDC DATA SHARING IN COMPUTER NETWORK PROJECT

The SDC project on Data Sharing in Computer Networks has been carried out in two phases. During the first phase, eight properties that are desired for sharing data in a computer network were defined. These properties are:

a. Users should be able to access any data in the network with one common language.

b. Data as they exist at each node should be sharable.

c. The DBMS currently used at each node to maintain the data should be utilized as the primary interfaces to the data.

d. Any approach to data sharing in a computer network should be evolutionary vs. revolutionary; i.e., a gradual shift in mode of operation vs. an abrupt change.

e. Fail-safe capabilities must be provided by any data sharing approach.

f. Implementation of a data sharing approach must be practical.

g. Effective system performance and user response times must be maintained.

h. The DBMS used in the network should be allowed to continue in their development so as to meet individual node requirements.

Given the foregoing properties as guidelines, five possible approaches to data sharing in computer networks were defined. These approaches were identified as centralized, integrated-central, integrated-distributed, data transformation, and standardized.

The centralized approach assumes that one service node would be responsible for all data management needs of the network. The integrated approach consists of providing access to the data management systems and local data at each node through the use of appropriate interfaces and a common user data language. The appropriate interface in this case is a language processor capable of translating user requests expressed from the common user data language into the data language supported by the DBMS of the node being accessed. These language translators can be implemented on a single

machine, the integrated-central approach, or at each node, the integrated-distributed approach. The data transformation approach involves providing the facilities needed to transform data (files or parts of files) from different nodes into a form that can be accessed by the DBMS local to the requesting user. The standardized approach consists of ensuring that the same set of data management services are implemented at each node in the network.

An evaluation of each of these approaches with respect to the properties desired for data sharing in computer networks is shown in Figure 5-1.

Based upon the results of this evaluation, SDC chose the integrated-central approach and began the second phase of its study on data sharing in computer networks. The work during this phase consisted of investigating a common user-oriented language, defining an intermediate language for expressing user requests functionally, and building two trial translator interfaces for translating the intermediate language ILDS into the data manipulation languages of two target systems.

The results of the SDC study can be stated as follows:

a. The integration of data management systems on a computer network is feasible through the use of common user data language and translation interfaces.

b. The translator interfaces were relatively easy to construct although it should be remembered that only a kernal version of the common user data language was implemented and translated.

c. The use of English as a common user data language is not as useful as might first be supposed.

d. Finally, it is possible to add functional power to the target DBMS by translating a single request in common user data language into a series of requests in the target data management language.

		Integrated			
	Centralized	Distributed	Central	Data Transfor-mation	Standardized
Existing DMSs and Data	No	Yes	Yes	Yes	No
Evolutionary/ Revolutionary	Rev.	Ev.	Ev.	Ev.	Rev.
Fail-Soft	No	Yes	No	No	Yes
Continuing Development of DMSs	No	Yes	Yes	Yes	No
Implementation (Relative)	Moderate	Moderate-Difficult	Moderate	Difficult	Difficult
Response-Time	Good	Moderate	Moderate	Slow	Good

Figure 5-1. Evaluation of Approaches to Data Sharing in Networks

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

5.2 THE MIT PROJECT ON COUPLING OF INTERACTIVE INFORMATION SYSTEMS

The purpose of this project was to investigate concepts and techniques in the areas of:

a. The virtual system concept by which the user perceives a computer network as a single homogeneous system.

b. A common command language synthesized from a basic language of primitive information retrieval functions.

c. A master index and thesaurus which stores the vocabularies of separate data bases along with index term interrelationships and counts.

d. A common bibliographic data structure in which the data elements for bibliographic information are hierarchically structured and interrelated among different data bases.

In addition to the theoretical study of the problem of sharing data on computer networks, an experimental interface was developed that connects the MEDLINE and Intrex retrieval systems via ARPANET communication links and that performs some of the networking functions of the virtual system.

Key results from this study were:

a. Interactive information retrieval systems can be coupled together effectively, even when they exist on separate networks, i.e., ARPANET and TYMNET.

b. A common user information retrieval language was defined that proved both easy to learn and use and was capable of representing all necessary commands in the target systems being accessed.

c. The demonstration of an online master index thesaurus which is a fundamental component for effective use of distributed data bases.

5.3 THE UDL PROJECT

In 1973, Logicon began an internal development project whose purpose was to identify the requirements for a standard user data language, define such a language, and determine how to most effectively implement a processor for this language. This project has been continued in conjunction with the COINS multi-language study and the remainder of this section describes the results of the UDL project.

Logicon Uniform Data Language (hereafter referred to as "UDL") has been designed to present a uniform user interface to a variety of data base management systems. The users of these data base management systems represent many areas of interest and therefore, pose a multitude of requests which must all be adequately satisfied through UDL and its associated software components. The data base management systems (hereafter referred to as "target systems") provide their own user languages to enable the user to interrogate, display, and update the data bases. These languages perform these functions but with such a variety of commands, command arguments, punctuation, and terminals, that it is virtually impossible for the users to become comfortable and proficient in communicating with as many as 10 target systems, which is the plan for the COINS Network.

Therefore, UDL is aimed at alleviating many of these problems which are generated by a network consisting of many languages and many data base management systems, specifically the COINS Network.

In preparation for the detailed design of UDL, several tasks were accomplished. One of these was the analysis of the data base management systems planned for the COINS Network. These target systems were examined with regard to the general areas of data structure and data manipulation. Another task was to develop a document which identified the requirements of the user,

the data language, the data base management system, and the network. And, finally, an investigation was performed of the current technologies in the field of language translation for data base systems.

The initial objectives to be met by UDL can be divided into two categories. In designing UDL as a pure language it must be

a. A single language.

b. Functionally complete and independent, and

c. An evolutionary language with respect to its functional power, i.e., simple and complex versions of the same function should be representable by the same basic root command, the difference being only in the presence or absence of optional parameters.

In designing UDL as a language which can be translated into other languages, it must

a. Be capable of accessing existing and potential target systems.

b. Present a uniform user interface.

c. Provide a consistent interpretation of data structures and commands as they are applied across the various target systems, and finally

d. Be constructed such that its data structure and commands can be mapped onto the target systems.

5.3.1 The Uniform Data Language

In describing UDL, its data structures must first be reviewed. There are five basic logical data structures in UDL:

a. Field

b. Aggregate

c. Record

d. File

e. Data base

A field is the smallest named data structure found in UDL. Fields are referenced by the user more often than any other data structure. There are single-valued fields, multi-valued fields, concatenated fields, and named multi-valued fields.

An aggregate contains collections of fields and/or other aggregates and is used primarily for forming hierarchical structures. Repeating groups and arrays are the two types of aggregates in UDL.

Records contain collections of fields and aggregates and is the second most important data structure in UDL since it is basically the unit of selectibility for almost all data base management systems. A record is distinguished by its type, i. e., all records with the same type must have the same fields and aggregates defined over them.

A file is a named collection of records containing one or more record types.

A collection of files is called a data base.

These UDL data structures provide the basis for the UDL commands. All target system data structures can be defined in the UDL data structure terminology and the target system data bases can be represented as UDL structured data bases.

UDL commands and statements are grouped into five categories:

a. Control Commands

b. Interrogation Statements

c. Display/Repeat Generation Statements

d. Update Statements

e. Data Manipulation Statements

Control commands are those which provide general system support to the user. A minimum set includes LOGON, LOGOFF, OPEN, and CLOSE. LOGON and LOGOFF allow the user access to the UDL processor. Associated with LOGON are the parameters user identification and password. LOGOFF has no parameters. To actually establish a connection with a target system the OPEN command is used. The user must specify as parameters to the OPEN command the name of each file he wishes to access and access information for that tile, such as a password. This will cause the UDL system to generate a log-on sequence to the target system and also an open file sequence if necessary. The CLOSE command removes this access to the target system file.

There is only one interrogation statement in UDL, the FIND statement. FIND selects records from a target system file based on specified parameters which are selection criteria and possibly the label of a previous FIND statement (indicating that the selection criteria are to be applied to a previously selected set of records instead of the whole file(s)). The selection criteria is composed of field-names, aggregate names and conditions to be met by the data items.

Report generation statements allow the user to obtain a simple display of what he has selected or at the other extreme to produce a report based on a complex display list and format statement. Before records can be displayed, they must be selected by a previous FIND statement. Then the DISPLAY statement can be used possibly in conjunction with the FORMAT statement, to generate the output. Other UDL statements associated with report generation are TABSET (establishes new user tab settings), ON (activates the output

of header and trailer lines), OFF (deactivates the output of header and trailer lines), HEADER (defines the format of a header line) and TRAILER (defines the format of a trailer line).

Update statements allow the UDL user to alter a file in several ways. The REMOVE statement removes entire records from a target system file; the records removed are those selected by a previous FIND statement. To add entire new records to a target system file, the user enters the CREATE statement along with a list of field-name/value pairs which describe the new record to be added. When an existing record is to be changed, the CHANGE statement will accomplish this. The parameters specified are field names of the fields to be changed and the new values for these fields. The actual records to be changed must have been selected by a previous FIND statement whose label is specified in the CHANGE statement. Two update statements are available to modify records in which there are occurrences of multi-valued fields and repeating groups. The ADD statement allows the user to add new field occurrences (under a repeating group) or new sub-field occurrences (in a multi-valued field). The new field name/value pairs are supplied as parameters to the ADD statement. The DELETE statement does the reverse of ADD; it is used to delete these occurrences. Both the ADD and DELETE statements are dependent on a previously selected set of records and therefore each of these statements must reference a FIND statement.

Most query languages associated with data base management systems supply some level of data manipulation, or data processing, capability. This feature is provided either as an integral part of the query language or as a language which is independent of the query language but one which can be interfaced to the query language to retrieve the records which are to be processed. The data manipulation capability in UDL is integral to the entire UDL language and

allows the user to intersperse other UDL statements and commands with the data manipulation statements. There are four general kinds of data manipulation statements:

a. Loop control

b. Execution control

c. Data processing

d. Declaration

Loop control statements involve the set-up and control of loops based on either a specified number, the number of records on a list, the number of occurrences of a specified data structure, or the number of unique values occurring in a specified field. Execution control statements include an IF-THEN-ELSE statement, a GO TO statement (both simple and computed), and a CALL statement (to transfer control to a procedure). There are several data processing statements available for list control and an assignment statement which assigns a value to a variable based on the evaluation of a numeric expression. Declaration statements allow the user to name and define a variable, and to name and define a procedure.

In summary, it is to be noted that in transforming UDL language command/statements to other languages, not all commands/statements must be translated. Some can be processed entirely in the UDL system. An example of statements which can be processed in the UDL system without translation are data manipulation statements. Entire records selected in the target system can be moved to the UDL system and then processed via data manipulation statements.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

5.3.2 The UDL Processor

The functional design for a UDL processor has been completed which will utilize a dual processor configuration of DEC PDP 11/45 computers. Detailed design will begin shortly. A production version of the UDL processor is scheduled for implemtnation on the PDP 11/70. A UDL processor is composed of:

a. A DEC PDP 11/45 or PDP 11/70 with two processors, main memory, mass storage, high speed communication lines, low speed communication lines for the terminals to be supported by the processor, a collection of terminals, and other peripherals.

b. A language processor for the uniform data language is capable of translating user requests to any one of several specified target system data languages.

c. A system executive which provides job monitoring, file, data base, and telecommunications management capabilities.

5.4 THE CCA DATACOMPUTER PROJECT

The Datacomputer is a large-scale data management and storage utility for use by a network of computers. The system is designed to provide facilities for data sharing among dissimilar machines, rapid access to large online files, storage economy through shared use of trillion-bit store and improved access control.

The Datacomputer was developed by the Computer Corporation of America under contract to ARPA. It is implemented on a DEC System-10. Key features provided by the Datacomputer include network data sharing services, large online files, access regulation, economy of scale, and specialization. All requests to the Datacomputer are stated in Datalanguage which is the command language of the system. Datalanguage contains facilities for data

description, for data base creation and maintenance, for selective retrieval of data, and for access to a variety of auxiliary facilities and services.

The Datacomputer concept is interesting from two standpoints with respect to the COINS multi-language problem. First, the Datacomputer provides an already existing method for establishing a centralized, standard data base management system node for COINS (see potential solutions 4 and 6 in Section 6). Second, the facilities provided by the Datacomputer, exclusive of the very large storage capacity, are the functions needed by any DBMS and it should be possible to construct a mini-Datacomputer using the Datacomputer design. A mini-Datacomputer would be an excellent candidate for use as the remote standard DBMS processor needed at each node in solution 7.

5.5 THE BELL TELEPHONE LABORATORIES XDMS PROJECT

In conventional data base management systems, all of the major software components — operating system, data base management system, and application programs — all execute on a single machine which has direct access to the data base on secondary storage. In 1971, Bell Telephone Laboratories began the XDMS project to determine if the data base management function could be placed on a dedicated back-end computer which accepts commands in a high level language from a host computer, accesses the data base on secondary storage, and returns the results to the host computer.

The advantages of back-end or remote DBMS processor are:

a. The hardware and software of the remote DBMS processor can be specialized for the required data base management functions providing more effective performance of these functions and relieving the host computer of the responsibility.

b. Data can be effectively shared between computer systems with different hardware if both systems are using a standard remote DBMS processor.

c. The data privacy and integrity capabilities of the remote DBMS processor are significantly better because it is dedicated to data management and can work with the host machine to identify and recover from operational errors and failures.

d. A full data base management system can be rapidly added to a new computer system with a remote DBMS processor by the construction of intracomputer interfaces which are simpler to develop than an entire DBMS.

The disadvantages of the remote DBMS processor are:

a. The cost of the second computer needed to implement the remote DBMS processor.

b. Possible mismatch in resource allocation through dedication of a processor entirely to the functions of data base management.

c. Possible response time problems due to the overhead imposed by the remote DBMS processor in contrast to a standard system.

Bell Telephone Laboratories' version of the remote DBMS processor, called the XDMS, was developed using a Digital Scientific META 4 computer. The cost of developing this processor was approximately $60, 000 and six man years of effort. The time required to implement the initial version of XDMS was 18 months.

This project was completed in 1973 and no further work has been done. However, the results of this work are extremely promising with the advantages achieved. The disadvantages can all be easily reconciled with minor changes in the system. Experimental results showed that the remote DBMS processor is an economically attractive alternative for data base management. Not only is there an apparent advantage in throughput per dollar, but there are also a number of new capabilities that such a configuration offers, such as the simultaneous sharing of a common data base by different computers and increased security of the data base.

5.6 THE IBM 3850 PROJECT

The IBM Corporation has developed a very large data storage system called the 3850 Mass Storage System which provide the hardware capabilities needed for any large centralized DBMS (see solutions 4 and 6 in Section 6). The major components of the 3850 Mass Storage System are the 3851 Mass Storage Facility, the 3830 Storage Control Unit, the 3333 Disk Storage and Control Unit, and 3330 Disk Storage Drives. Storage capacity of the system ranges from 35 billion to 472 billion bytes of information. Data is stored in direct access storage format on data cartridges. The data cartridges are contained in the 3851 Mass Storage Facility. Each data cartridge contains a wide spool of magnetic tape 770 inches long and capable of containing 50 million bytes of data.

Whenever data is needed by the system for processing or updating, the 3850 Mass Storage System transfers it from the data cartridges to the 3330 Disk Storage System under the control of the 3830 Storage Control Unit and the 3333 Disk Storage and Control Unit by means of a process called staging. Once the data has been staged onto disk storage, it can be accessed by the DBMS of the system just like any other file.

SECTION 6

DESCRIPTION OF POTENTIAL COINS SOLUTIONS

Eleven potential solutions have been developed for the COINS multi-language problem during the course of this study. These solutions are based upon the results of the previously described requirements, DBMS, and advanced technology analyses coupled with a thorough examination of current and planned COINS operations. The purpose of this section is to provide a short but precise description for each of the 10 potential COINS solutions.

Each of the potential COINS solution descriptions consists of:

a. A basic solution definition which indicates in one or two sentences the overall solution strategy.

b. A diagram of a representative node configuration utilizing the solution and a discussion of practical alternative configurations.

c. The operation of the representative node when employing the potential solution.

d. The tasks, schedules, and costs associated with effecting the solution.

e. An evaluation of the solution with respect to user, node, and COINS requirement.

f. A summary of the overall advantages and disadvantages of the solution.

The 10 potential solutions for the COINS multi-language problem to be described in the remainder of this section are summarized as follows:

Solution	Basic Definition
1	Select and install standard hardware, DBMS, and user data language processors (UDLP) at each node.
2	Use a selected set of current node hardware and secure and install a standard DBMS and UDLP.
3	Use current node hardware and DBMS but provide a centralized UDLP to translate from a standard COINS user language into data language of a target node.
4	Use current node hardware to support node user access to centralized UDLP and DBMS.
5	Use current node hardware and DBMS for data management functions but install distributed UDLP at each node to provide user access and a standard COINS user data language.
6	Use current hardware for file creation and maintenance, move files shared by COINS to central DBMS system, and provide access to users through distributed UDLP.
7	Use current hardware and DBMS but augment DBMS with dedicated special DBMS processor and provide access to users through distributed UDLP.
8	Make no hardware or software changes and establish user support centers at each node consisting of trained personnel called data specialists, who will serve as users' interface to COINS.
9	Combination of solutions 2 and 5.
10	Combination of solutions 2 and 7.

6.1 SOLUTION 1: STANDARD HARDWARE, DBMS, AND UDLP

Solution 1 consists of selecting and installing standard hardware, DBMS, and user data language processors (UDLPs) at each node served by COINS.

6.1.1 Solution 1 Configuration

The basic configuration for solution 1 is shown in Figure 6-1.

Two primary alternatives exist for configuring a node using solution 1: secure all components from a single supplier or from multiple suppliers. In the case of single supplier, three manufacturers have systems which appear to be most effective for configuring solution 1: Honeywell, IBM, and Univac. The Honeywell 6880 system using the MULTICS operating system has many access control and data base management capabilities needed by a COINS node. The IBM 370 series are used at four of the eight current and potential nodes for COINS and has the advantage of 3850 technology. The Univac 1100 series

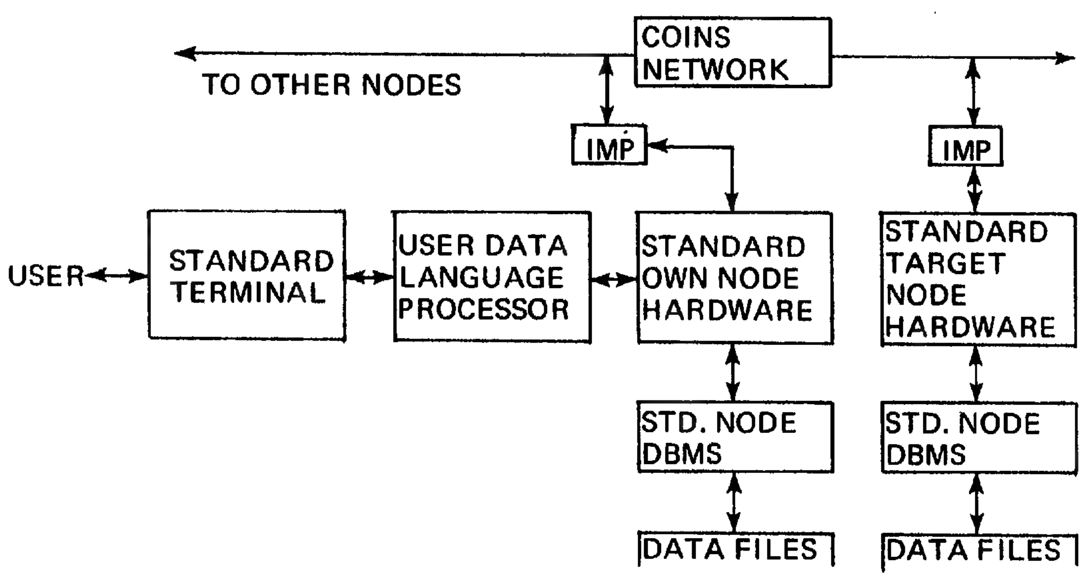

Figure 6-1. Solution 1 Configuration

is used at two of the current and potential nodes for COINS and has an operational version of the CODASYL DBTG DBMS. In the case of multiple suppliers, the hardware manufacturers listed above remain the same, but DBMS can be secured from Computer Corporation of America, Management Research Inc., TRW, Cullinane, etc. which are capable of meeting a node's DBMS requirements. The user data language needed to support COINS users does not exist on any currently available hardware. The primary deficiencies are in the areas of user access control and user support with respect to COINS file guides and command utilization. The UDLP can be produced by one of the COINS nodes on behalf of the other nodes and shared with them, or it can be constructed by either the hardware supplier or an outside contractor. The terminals available to each node will be standard.

6.1.2 Solution 1 Operation

The operation of a solution 1 node in COINS is straightforward. A user wishing to query a COINS file enters his request via his terminal into his host node system where it is processed by the host node UDLP. This processing consists of determining whether this user has access to the file being queried and, if so, routing the query to the node containing the desired file. Upon resolution of the query by the target node, a response is sent back to the user. No translation is required for either the query or response because of the standard hardware and software being used at each node. (If this solution were adopted for COINS, careful consideration must be given to establishing and maintaining COINS node standards to ensure that node-unique applications do not destandardize the node for COINS.)

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

6.1.3 Solution 1 Tasks, Schedules, and Costs

The primary tasks to be accomplished to implement solution 1 consist of:

a. Establish node system hardware, DBMS, and UDLP requirements and select standard hardware, DBMS, and UDLP.

b. Install standard hardware and DBMS at each node.

c. Define and develop UDLP needed to support COINS users.

d. Convert individual node programs and files.

e. Integrate nodes and test overall operation of COINS II.

The complexity involved in executing and coordinating these tasks is well understood based upon previous experience in the MULTICS and WWMCCS programs.

The schedules and costs associated with these tasks are shown in Figure 6-2.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. SELECT HARDWARE AND DBMS						900
2. INSTALL DBMS AND HARDWARE						13,527
3. DEVELOP USER DATA LANGUAGE PROCESSOR						480
4. CONVERT FILES AND PROGRAMS						5,536
5. INTEGRATE NODES AND TEST COINS II						2,350
TOTAL						22,793

Figure 6-2. Costs and Schedules for Solution 1

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

Additional cost estimates indicating costs by component and node are shown in Figure 6-3. Costs are shown in thousands of dollars.

6.1.4 Solution 1 Evaluation

An evaluation of the capabilities afforded to the COINS user, node, and network is indicated on the following page.

ITEM	DIA	NPIC	NSA(SOLIS)	NSA(TILE)
1 HARDWARE — UNIVAC 1100	4, 509	—	4, 170	4, 674
2 DBMS — DMS 1100	—	—	—	—
3 UDLP	200	200	200	200
4 FILE CONVERT	210	30	30	240
5 PROGRAM CONVERT	1, 000*	—	256	1, 000*
TOTALS	$5, 919	$230	$4, 656	$6, 114

*Costs shown here are to indicate magnitude of costs for each node. Univac 1110 used for costing and program conversion costs at both DIA and NSA (TILE) are higher than the $1 million shown. The $1 million represents our assessment of costs that should be assigned to COINS as the result of a change in hardware at these nodes. Program conversion costs at DIA are estimated at $4. 3 million and $3. 0 million at NSA for TILE.

Figure 6-3. Solution 1 Costs by Component and Node

	Capabilities	Evaluation
1	User support	With a well-defined and developed UDLP, adequate user support should be available to all user types.
2	Data language	The single language currently available for any of the practical hardware choices is limited with respect to structure, dimension, and extension.
3	Data structures	CODASYL DBTG-based DBMS support network structures; the remainder are limited to hierarchical. This may not be a problem: all COINS files are hierarchical.
4	Data management functions	Complete complement of data manipulation functions available; however, not all in interactive mode.
5	Uniform data management at node	Yes, given that effective standards can be established and maintained at the nodes.
6	Minimum impact on node	The impact varies depending upon which set of hardware is selected as standard. Selecting Univac 1100 places less impact on current nodes, but potential nodes are primarily IBM.
7	Terminal compatibility	Yes, again a community or network standards problem.
8	Privacy and integrity	The Honeywell 6880 with the MULTICS operating system is only hardware close to capable, but with standard hardware privacy and integrity of data more likely.
9	Network compatibility	Yes, most useful solution for COINS in that only one node-IMP interface required.
10	COINS schedule	Meets COINS schedule.
11	Performance	Given proper configuration and balanced use of resources by node and COINS, performance satisfactory.
12	Cost	Depending upon hardware selected, this is most expensive solution.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

An evaluation of the potential problems which could be faced if solution 1 is selected follows:

	Potential Problem	Evaluation
1	GSA constraints against single vendor	Constraint has been overcome in past but little chance in this case.
2	Constrained to single architecture and technology	This is a serious problem and may impinge upon node mission.
3	Conversion to next generation must be in parallel	Probable with high drain on already inadequate personnel supply.
4	Delay in standardization due to slow conversion	Probable due to lack of personnel and current node work loads.

6.1.5 Solution 1 Advantages/Disadvantages

	Advantages		Disadvantages
1	Single user language	1	Excessive node impact
2	Standard DBMS	2	Single vendor unlikely
3	Standard terminals	3	Bound to one architecture and technology
4	Uniform network/node interfaces	4	Next generation conversion massive
5	File and program sharing	5	Slow conversion likely
6	Possible load sharing	6	Most costly solution

6.2 SOLUTION 2: SELECTED HARDWARE, STANDARD DBMS, AND UDLP

Solution 2 consists of using a selected set of current and potential node hardware and securing and installing a standard DBMS and UDLP.

6.2.1 Solution 2 Configuration

The basic configuration for solution 2 is shown in Figure 6-4.

All nodes will use one of four selected hardware system types: Control Data 6/7000 series, Honeywell 6000 series, IBM 370/155-165 or IBM 370/158-168 series, and the Univac 1100 series. There are two alternatives for configuring the standard DBMS. The first alternative involves selecting a standard DBMS which operates or can be made to operate on the selected hardware systems.

SYSTEM 2000 of MRI Systems Corporation is an example of this type of DBMS. SYSTEM 2000 operates on CDC 6/7000 series, IBM 370, and Univac 1100 series. MRI has indicated its willingness to produce a Honeywell 6000

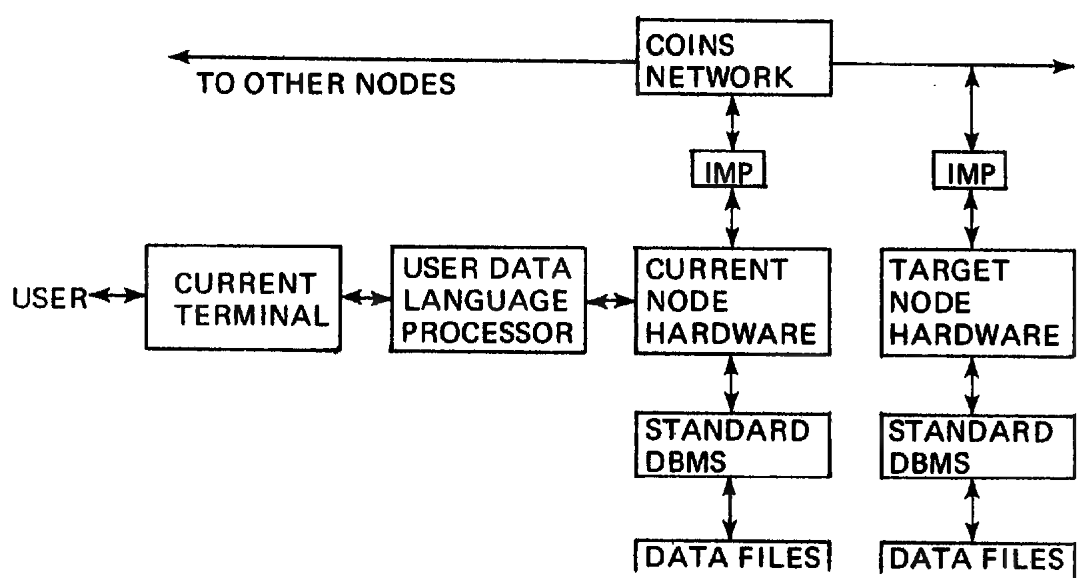

Figure 6-4. Solution 2 Basic Configuration

version for $300 thousand in 18 months. The other alternative is to select a standard DBMS such as the CODASYL DBTG-based systems such as DMS-1100 for the Univac 1100 or Cullinane's IDMS for the IBM 360/370; Honeywell, Burroughs, and CDC are all producing their own versions of the same DBMS.

The UDLP that is needed for the first alternative, while on different hardware, is logically the same. The UDLP that must be constructed if the second alternative is selected (i. e., multiple vendors for the standard DBMS) is essentially unique for each node with different hardware.

The terminals available to the user at each node are standard with respect to that node but may not be compatible with terminals used at other nodes.

6.2.2 Solution 2 Operation

Assuming the use of standard DBMS supplied by a single vendor, the operation of a typical node under solution 2 is as follows. The user enters his data request (interrogation, updating, manipulation, or display) through his terminal. The UDLP on his host node determines his access rights and the location of the desired file. If the file is on the host node, the host node version of the standard DBMS is invoked and, upon completion of the request, provides a response to the host node UDLP. If the file being accessed is on another node, the request is transmitted to that node where the request is treated exactly the same as on the host node except for reconciliation of minor differences in the operations of the supporting hardware systems (hardware, operating system, etc.).

6.2.3 Solution 2 Tasks, Schedules, and Costs

The primary tasks to be accomplished in implementing solution 2 include:

a. Identify selected set of hardware systems to be used and procure standard DBMS.

b. Construct UDLP for each node and integrate with version of standard DBMS at that node.

c. Install standard DBMS and UDLP at NPIC and convert NPIC files.

d. Install standard DBMS and UDLP at NSA for SOLIS and TILE and convert SOLIS and TILE files.

e. Install standard DBMS and UDLP at DIA and convert DIA files.

The schedules and costs associated with these tasks are shown in Figure 6-5.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. SELECT STD DBMS						480
2. DEVELOP USER DATA LANGUAGE PROCESSOR FOR EACH NODE						576
3. INSTALL STD DBMS AT NPIC AND MOVE FILES						150
4. INSTALL STD DBMS AT NSA						1,000
5. MOVE SOLIS AND TILE FILES						240
6. INSTALL STD DBMS AT DIA AND CONVERT DIAOLS FILES						1,210
TOTAL						3,656

Figure 6-5. Costs and Schedules for Solution 2

6.2.4 Solution 2 Evaluation

An evaluation of the capabilities afforded to the COINS user, node, and network is indicated in the following.

	Capability	Evaluation
1	User support	Possible to provide minimal user support if UDLP can be developed to single interface; i.e., SYSTEM 2000 or moved to front-end processor.
2	Data language	The user data language currently available with SYSTEM 2000 or DBTG version DBMS is not acceptable with respect to structure and dimension.
3	Data structures	SYSTEM 2000 provides hierarchical with limited inversion. DBTG versions of DBMS provide hierarchical or network.
4	Data management functions	Complete complement of data management functions available; not necessarily compatible in interactive and batch modes.
5	Uniform data management at node	Yes, at the logical level only. Physical storage structures determined by hardware and operating system.
6	Minimum impact on node	This solution provides relatively low impact on each of the current nodes but high impact on potential COINS nodes.
7	Terminal compatibility	Terminals vary by hardware system at node.
8	Privacy and integrity	Solution 2 helps at least in that standards of data management are established at each node.
9	Network compatibility	With selected hardware set, limited standardization occurs.
10	COINS schedule	Best solution with respect to schedule.

	Capability	Evaluation
11	Performance	Will vary with node.
12	Cost	This solution has lowest cost.

An evaluation of the potential problems which could be faced if solution 2 is selected follows:

	Potential Problem	Evaluation
1	Failure of single sofware source	Possible but not likely and inexpensive offsetting actions available.
2	Maintenance and upgrade costs because of multiple hardware types	Likely problem, but current situation worse in that logical models for COINS DBMS are not equivalent.
3	Exclusion of certain hardware types because of standard DBMS development costs	Not a real problem if application different hardware worth more than $300, 000 to retain.
4	Any standard DBMS will require extension to meet COINS requirements	Yes, but extensions can be staged with minimal impact on production.

6.2.5 Solution 2 Advantages/Disadvantages

	Advantages		Disadvantages
1	Single user language	1	May require DBMS development for DIA
2	Standard data definitions for all COINS files	2	Possible reliance on single vendor
3	Standard logical data structures with result that command semantics identical at each node	3	UDLP development
4	Minimal hardware changes	4	File and application conversion

Advantages		Disadvantages
	5	Multiple terminal types
	6	Nonstandard physical storage structures

6.3 SOLUTION 3: CURRENT HARDWARE AND DBMS, CENTRAL UDLP

Solution 3 utilizes current node hardware and DBMS but provides for a standard COINS user data language through a centralized UDLP which translates the standard user data language into a specified target data language and any responses back into the standard user data language.

6.3.1 Solution 3 Configuration

The basic configuration for solution 3 is shown in Figure 6-6.

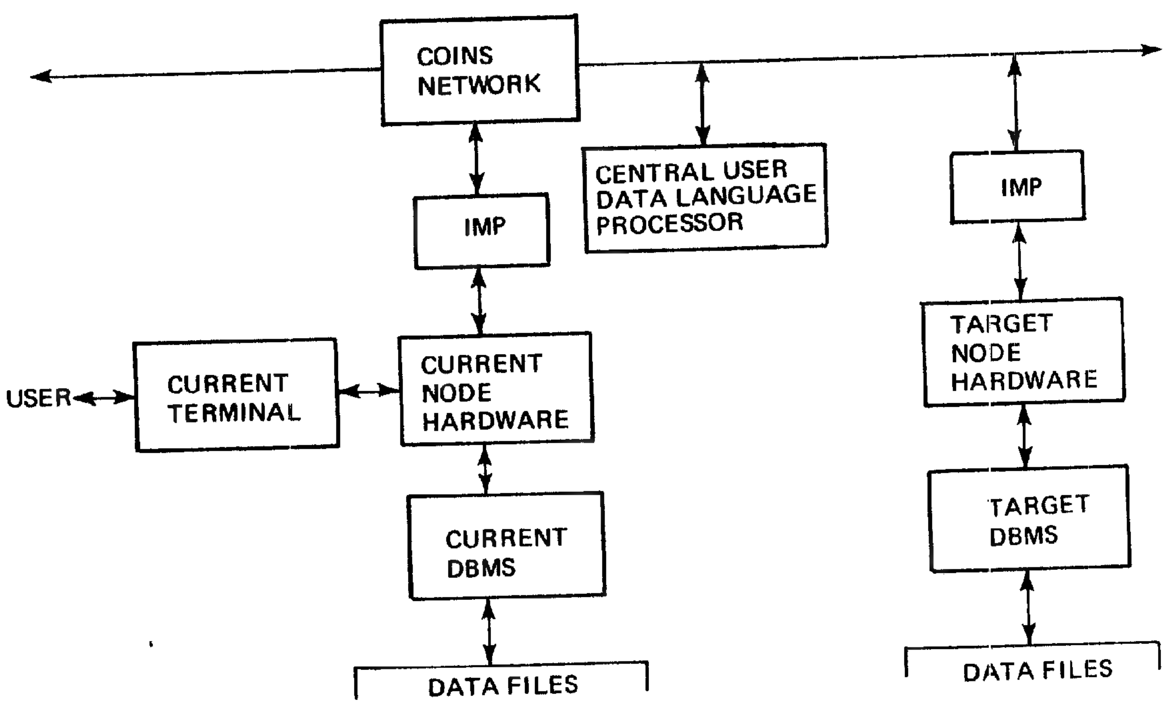

Figure 6-6. Solution 3 Basic Configuration

In the solution 3 configuration, all nodes retain their current hardware and DBMS. A processor is added to each node to manage transactions between users of that node and a new network node which is a central UDLP. Two configuration alternatives exist for this solution. In the first case, the processor to handle user requests can be a software module in each host node system. In the second case, the user request processor can be implemented as front-end processors for each host node. The latter alternative has the advantage of providing a uniform user interface with respect to terminals and user support functions. An examination of current COINS nodes indicates that user request processors, hereafter called uniform terminal processors, are needed either because the current node capacity will be exceeded or a hardware change is anticipated. An examination of potential COINS nodes produced similar results. Therefore, implementation and installation of uniform terminal processors (UTPs) are recommended for solution 3.

6.3.2 Solution 3 Operation

A COINS user enters his data transaction (retrieval, update, manipulation, or display request) through the standard terminal provided by the use of UTPs at each node. The UTP validates the user's access rights, determines the routing for the transaction, edits the transaction for any processing by the host node, and passes the transaction to the host node. The host node, upon receipt of the user transaction from the UTP, accesses COINS and transmits the request to the central UDLP. The user data language processor determines the target system to be accessed by the transaction and translates the user data language into the appropriate target data language. The user's transaction, now expressed in the required target data language, is transmitted to that target node for processing. When the transaction has been processed, the response to the user's transaction is transmitted to the

UDLP and then back to the host node and the originating user. It should be noted that in this approach, even though the transaction is for the user's own host node, the transaction must first be sent to the UDLP and the response back to the UDLP prior to finally being received by the user.

6.3.3 Solution 3 Tasks, Schedules and Costs

The tasks needed to implement solution 3 are:

a. Defining a standard user data language for the COINS users.

b. Developing a central user data language processor.

c. Developing the uniform terminal processors.

d. Installing and integrating the uniform terminal processors at each user node.

e. Installing and integrating the user data language processor as a node in COINS.

f. Testing full network operation on a node-by-node basis.

The schedules and costs associated with performing these tasks are shown in Figure 6-7.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE USER DATA LANGUAGE						120
2. DEVELOP CENTRAL USER DATA LANGUAGE PROCESSORS UDLP(S)						576
3. DEVELOP UNIFORM TERM PROCESSORS						180
4. INSTALL AND INTEG UTP(S)						6,964
5. INSTALL AND INTEG UDLP(S)						6,146
6. TEST FULL NETWORK						120
TOTAL						14,096

Figure 6-7. Costs and Schedules for Solution 3

6.3.4 Solution 3 Evaluation

An evaluation of the capabilities afforded to the COINS user, node, and network by solution 3 follows:

	Capability	Evaluation
1	User support	Given the use of uniform terminal processors and a single user language, solution 3 provides good user support.
2	Data language	If a single user data language is defined that is equivalent to the Logicon uniform data language, the data language requirement can be satisfied.
3	Data structures	The data structures seen by the user will be those provided through the user data language, which mirror the data structures available in COINS.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

	Capability	Evaluation
4	Data management	The data management functions available to the user are those defined by the user data language and represented by the nodes.
5	Uniform data management at node	No, each node retains its own DBMS.
6	Minimum impact on node	The primary impact on each node results from installation and integration of the uniform terminal processors.
7	Terminal compatibility	The uniform terminal processors at each node allow terminal standards to be invoked for COINS.
8	Privacy and integrity	The use of access control procedures at the uniform terminal processor, within the host node, at the user data language processor, and at the target node provide excellent data privacy and integrity.
9	Network compatibility	Provides good network compatibility with single interface needed between user terminal processors and nodes and between nodes and user data language processor.
10	COINS schedule	Probably delays effective network operations by as much as one year.
11	Performance	Probable serious bottleneck in central user data language processor and possible bottleneck at single nodes.
12	Cost	Moderate cost, $14, 096 per terminal for a thousand terminals served.

An evaluation of the potential problems associated with adopting this solution follows:

	Potential Problem	Evaluation
1	Performance or reliability due to central user data language processors	The probability of poor performance with this solution is very high. In addition, even with redundant processors to offset the reliability problem, sluggish performance is likely.
2	User data language not able to represent all needed functions and knowledge of node sublanguages required by users	Logicon's uniform data language project has shown that a user data language can be designed where knowledge of node sublanguages is not required.

6.3.5 Solution 3 Advantages/Disadvantages

Advantages

1. Single user language
2. Minimal hardware and DBMS changes needed at each node
3. Standard terminals at each node if uniform terminal processors used
4. No file conversions are needed
5. Extended user support involving command prompting and online COINS file guides available with uniform terminal processors

Disadvantages

1. Potential performance problems at central user data language processors
2. Solution involves user data language processor and uniform terminal processor development
3. Multiple logical and physical interfaces are required because each node hardware may be different

6.4 SOLUTION 4: CURRENT HARDWARE, CENTRAL DBMS AND UDLP

Solution 4 consists of using current node hardware to support node user access to a centralized user data language processor and DBMS.

6.4.1 Solution 4 Configuration

The basic configuration for solution 4 is illustrated in Figure 6-8. In the solution 4 configuration, all nodes retain their current hardware. A processor is added at each node to manage transactions between the users at that node and a new network node (probably nodes to provide redundancy both in function and location) consisting of centralized user data language processors and DBMS. Two options exist for the solution 4 configuration with respect to the user transaction processors. In the first option a uniform terminal processor can be added to the node as a software program. The second option involves construction of a user terminal processor using minicomputers and installing one or more as front-end processors at each node. Again, based upon an analysis of the current and potential nodes for COINS, the construction and installation of uniform terminal processors as front-end processors is recommended.

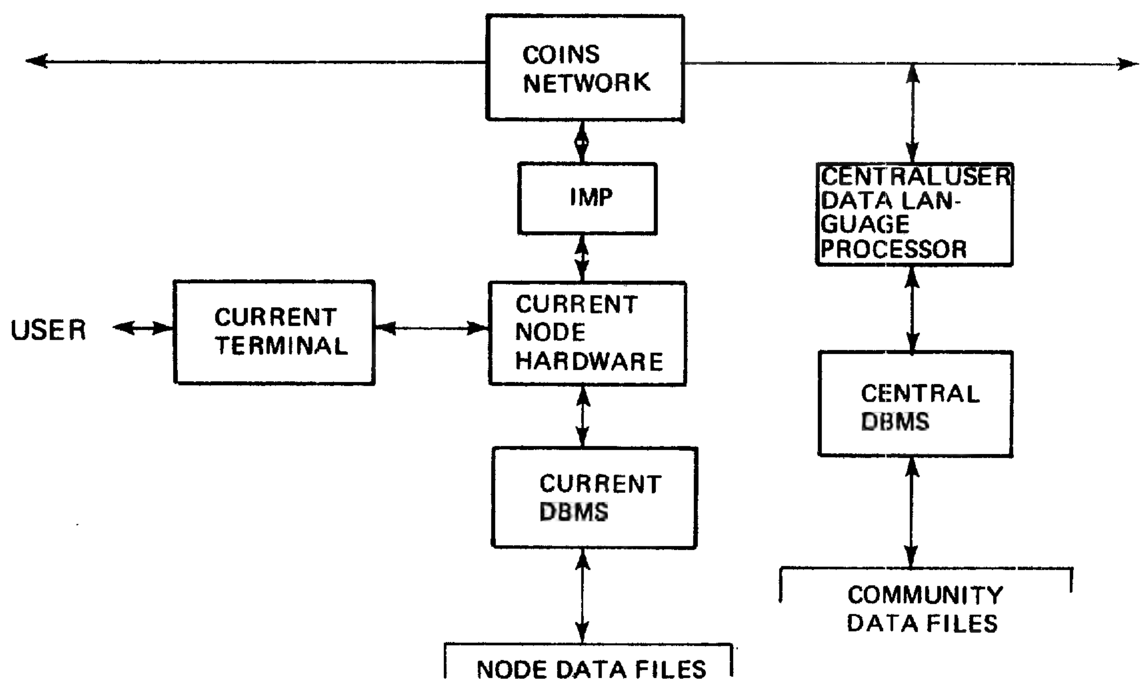

Figure 6-8. Solution 4 Basic Configuration

6.4.2 Solution 4 Operation

Utilizing solution 4, the operation of a typical node in COINS is as follows: a user wishing to access a COINS file, enters his request through his terminal connected to the user transaction processor. The user transaction processor validates the user access rights, applies the proper routing to the request using processor contained COINS file directors, and transfers the request to the host node system. The host node system notes the type of request, establishes a transaction response program on behalf of the user and sends the request on to the central user data language processor and DBMS nodes. When the transaction has been processed by the central UDLP/DBMS node, a response is sent back to the originating node. If the response involves significant amounts of data, the transaction response program established by the host node handles the response. Otherwise, the response is transferred directly to the user by the host node via the user transaction processor. The user transaction processor is equivalent to the uniform terminal processor described previously.

6.4.3 Solution 4 Tasks, Schedules, and Costs

The primary tasks to be accomplished to implement solution 4 include:

a. Define user data language to satisfy COINS user requirements.

b. Develop user data language processor for use at central nodes.

c. Develop uniform terminal processors (UTP).

d. Develop COINS centralized DBMS.

e. Install user data language processors, uniform terminal processors and centralized DBMS.

f. Test each node and then full network operation.

The schedules and costs associated with performing these tasks are shown in Figure 6-9.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE USER DATA LANGUAGE						120
2. DEVELOP USER DATA LANGUAGE PROCESSOR (UDLP)						576
3. DEVELOP UNIFORM TERM PROCESSOR						180
4. DEVELOP COINS CENTRAL DBMS						810
5. INSTALL UDLP, UTP, AND DBMS						13,220
6. TEST EACH NODE AND FULL NETWORK						842
TOTAL						16,048

Figure 6-9. Schedules and Costs for Solution 4

6.4.4 Solution 4 Evaluation

The capabilities and potential problems associated with solution 4 were evaluated and the results follow:

Capabilities	Evaluation
1 User support	Given the use of uniform terminal processors, solution 4 has the capabilities to serve each of the COINS user types effectively. The UTP provides highly interactive response, file guide assistence, command utilization support, and effective access control. The host node provides effective processing resources.
2 Data language	The data language requirements can be satisfied with a single user data language equivalent to the Logicon uniform data language.

Capabilities	Evaluation
3 Data structures	The data structures that are supported by solution 4 are a function of the COINS DBMS that is developed for the central UDLP/DBMS nodes. COINS files are primarily hierarchical with inverted lists but there are a number of applications which require data structures so a CODASYL DBTG version DBMS should be considered.
4 Data management functions	The data management functions available to the user are complete at the central DBMS but will vary at each user node. It is possible for the user under solution 4 to accomplish all of his data management functions at the central DBMS nodes but it may be more convenient to use the local node for output resulting in increased network data transmission and local node processing loads.
5 Uniform data management at nodes	Since the data management for COINS files is being handled primarily at the central nodes, uniform data management is almost achieved under solution 4. Since the owners of the COINS files exist at the local nodes and will maintain the master version of their files at the local node and when the local output mentioned under data management functions are considered, solution 4 has some deficiencies with respect to uniform data management.
6 Minimum impact at node	The primary impact on each node under solution 4 results from the installation and integration of the uniform terminal processors and the need to maintain two sets of files, one at the central nodes and a private version at the node.
7 Terminal compatibility	The uniform terminal processors at each node allow terminal standards to be established and maintained on COINS. The use of standard terminals will aid in the training and support of the COINS user community.
8 Privacy and integrity	Excellent data privacy and integrity capabilities available with solution 4.

	Capabilities	Evaluation
9	Network compatibility	Provides good network compatibility with single interface needed between uniform terminal processors and nodes and between nodes and user language processor/DBMS nodes.
10	COINS schedule	Probably delays effective network operations by as much as one year.
11	Performance	Possible serious bottleneck in central user data language processor/DBMS. Further evaluation is required.
12	Cost	Moderate cost, average cost per terminal is $15,748 per terminal for a thousand terminals served.

	Potential Problems	Evaluation
1	Possible performance problem due to centralized UDLP/DBMS	Based upon current COINS activity, it is probable that occasional slow response times will occur. If COINS activity continues to increase which is likely, solution 4 performance may not be satisfactory for significant portions of the day.
2	Duplicate files and nonsupport of central files by file sponsors	Duplicate files will almost certainly exist and the current trend is for file sponsors to maintain files for their own agency most effectively.
3	Vulnerability of centralized UDLP/DBMS nodes to catastrophe	Duplicate UDLP/DBMS nodes will solve this problem but it raises the cost of the solution and increases the file maintenance problem.

6.4.5 Solution 4 Advantages/Disadvantages

Advantages	Disadvantages
1 Single user data language	1 Probable performance problems.
2 Little or no change to current nodes	2 Duplicate files must be maintained which defeats the purpose of COINS.
3 Centralized standard DBMS	3 Development effort required for user data language processor, COINS DBMS and uniform terminal processors.
	4 File conversion and maintenance required to move current files to centralized DBMS.

6.5 SOLUTION 5: CURRENT HARDWARE AND DBMS, REMOTE UDLP

Solution 5 consists of using current node hardware and DBMS for the data management functions but installing distributed user data language processors (UDLP) at each node to provide user access and a standard COINS user data language.

6.5.1 Solution 5 Configuration

The basic configuration for solution 5 is shown in Figure 6-10. Several alternative configurations are possible for this solution. The first alternative involves implementing the user data language processor as a software module on each host node. While this approach limits the amount of additional hardware required, multiple hardware dependent versions of the user data language processor must be developed. The second alternative involves constructing the user data language processors as front-end processors for each host node. This alternative, while reducing the number of IMPs needed and providing for standard user data language processors (software and hardware), causes an additional load to be placed on the host node. Based

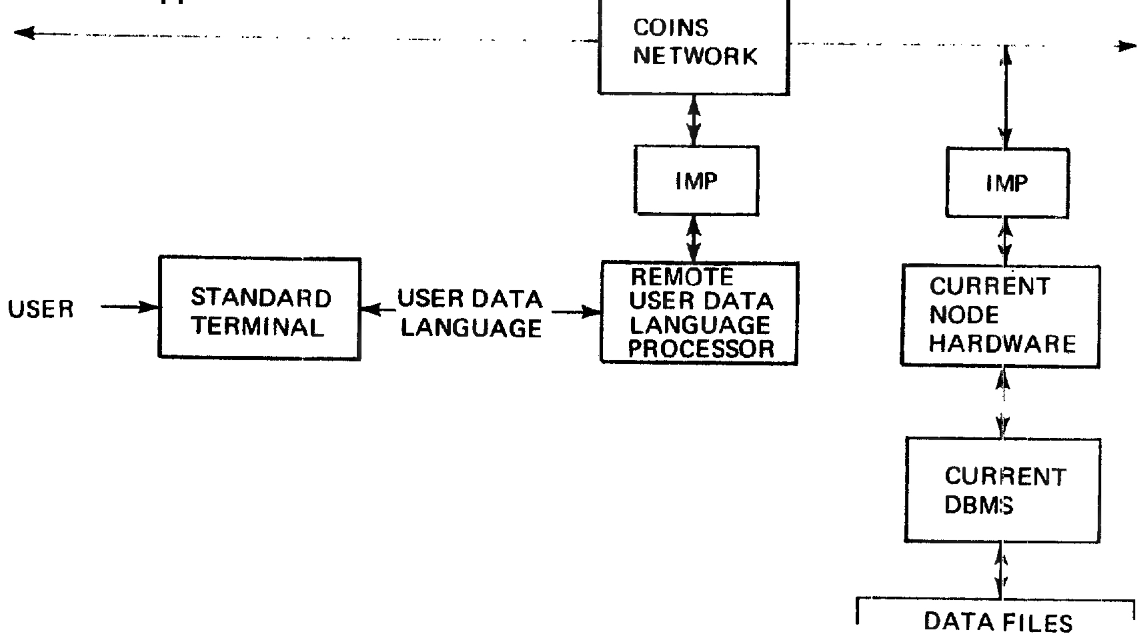

Figure 6-10. Solution 5 Basic Configuration

upon our evaluation, the best alternative is the configuration shown in Figure 6-10. This configuration resolves the problems of the previous two alternatives and provides the advantage of allowing the node users to access COINS files even if their own host node goes down.

The user data language needed for this solution must also be the uniform data language designed by Logicon or its equivalent.

6.5.2 Solution 5 Operation

The operation of a user data language processor terminal is quite an improvement for the user over some of the other solutions. The user enters his request in the user data language, regardless of which target system he is addressing. The request is processed in the user data language processor where it is translated into the language of the target system, and also

reformatted to overcome any terminal incompatibilities between the standard terminal of the UDLP and the terminal type supported by the target system. The request is then formatted for transmission over the COINS network, and finally transmitted to the target system. When a response is received from the target system, the response is interpreted and formatted for transmission to the user as standard response in the user data language.

6.5.3 Solution 5 Tasks, Schedules and Costs

The primary tasks to be accomplished in order to implement solution 5 include:

a. Define the user data language to be used for the COINS network, utilizing the uniform data language as a base and adding any additional features as required.

b. Develop the user data language processors for each node.

c. Develop a node interface in each target system to interface to the network and manage user transactions against the files maintained by that node.

d. Install and integrate the user data language processors required for each node.

e. Integrate the total network and test the overall operation of COINS II.

The schedules and costs associated with performing these tasks are shown in Figure 6-11.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE USER DATA LANGUAGE						120
2. DEVELOP USER DATA LANGUAGE PROCESSOR (UDLP)						576
3. DEVELOP NODE INTERFACES						600
4. INSTALL AND INTEG UDLP(S) AND NODES						8,436
5. INTEGRATE AND TEST NETWORK						180
TOTAL						9,912

Figure 6-11. Schedules and Costs for Solution 5

6.5.4 Solution 5 Evaluation

An evaluation of the capabilities and potential problems associated with solution 5 follows:

Capabilities	Evaluation
1 User support	Since free-standing UDLPs are used, solution 5 has the capabilities to serve the COINS user types more effectively than any other solution other than solutions 6 and 7. The UDLP provides highly interactive response, file guide assistance, command usage support, effective access control, and a standard user data language.
2 Data language	The data language requirements are completely satisfied with a single user data language like the Logicon uniform data language.

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

	Capabilities	Evaluation
3	Data structures	The data structures supported by solution 5 are those defined by the user data language and supported by the current COINS DBMS. Current COINS DBMS are hierarchical with inverted lists. NPIC is converting to DMS-1100 which provides network data structures, but is not planning to use them immediately.
4	Data management functions	The data management functions available to the user are complete as defined by the user data language. Because each node data management capabilities vary, the standard user data language used by COINS will have been translated into multiple target data language statements in some instances to ensure uniform and complete data management functions.
5	Uniform data management at nodes	Solution 5 provides a uniform data interface for the user, but each node will continue to have its own unique data management characteristics.
6	Minimum impact at node	The primary impact on each node under solution 5 will result from installation and integration of the user data language processors and development of the node interface modules.
7	Terminal compatibility	User data language processors at each node will provide standard terminals for each COINS user.
8	Privacy and integrity	Two levels of access control are provided by solution 5, one at the user data language processor and the second at the target mode. As a result, data privacy characteristics of solution 5 are excellent but data integrity is a node dependent function.
9	Network compatibility	Provides good network compatibility with single interface required between UDLP and IMPs and betweem IMPs and target nodes.

Capabilities	Evaluation
10 COINS schedule	Probably delays effective operation of COINS for one year.
11 Performance	Performance should be excellent unless target node loading occurs as currently happens with DIAOLS.
12 Cost	Next to solution 2, this is the lowest cost solution.

Potential Problems	Evaluation
1 Possible loading at target node	Not a significant problem now and it can be overcome with improved target node hardware and scheduling procedures.

6.5.5 Solution 5 Advantages/Disadvantages

Advantages	Disadvantages
1 Single user language	1 User data language processor development is required.
2 Little or no changes at host nodes	2 Potential overloading at specific node.
3 Users are able to access remainder of COINS files even if own node is down.	3 User data language processor must be expanded each time a node is added to the network.
4 Standard terminals for COINS users	
5 Extended user support if available in UDLP.	

6.6 SOLUTION 6: CURRENT HARDWARE, CENTRAL DBMS, AND REMOTE UDLP

Solution 6 consists of using current hardware for file creation and maintenance, move the files shared by COINS to central DBMS node, and provide access to these files through distributed user data language processors.

6.6.1 Solution 6 Configuration

The basic configuration for solution 6 is shown in Figure 6-12. Solution 6 has the same configuration alternatives with respect to placement of the UDLP as solution 5 and the free-standing IMP connection for the UDLP is recommended for solution 6 also.

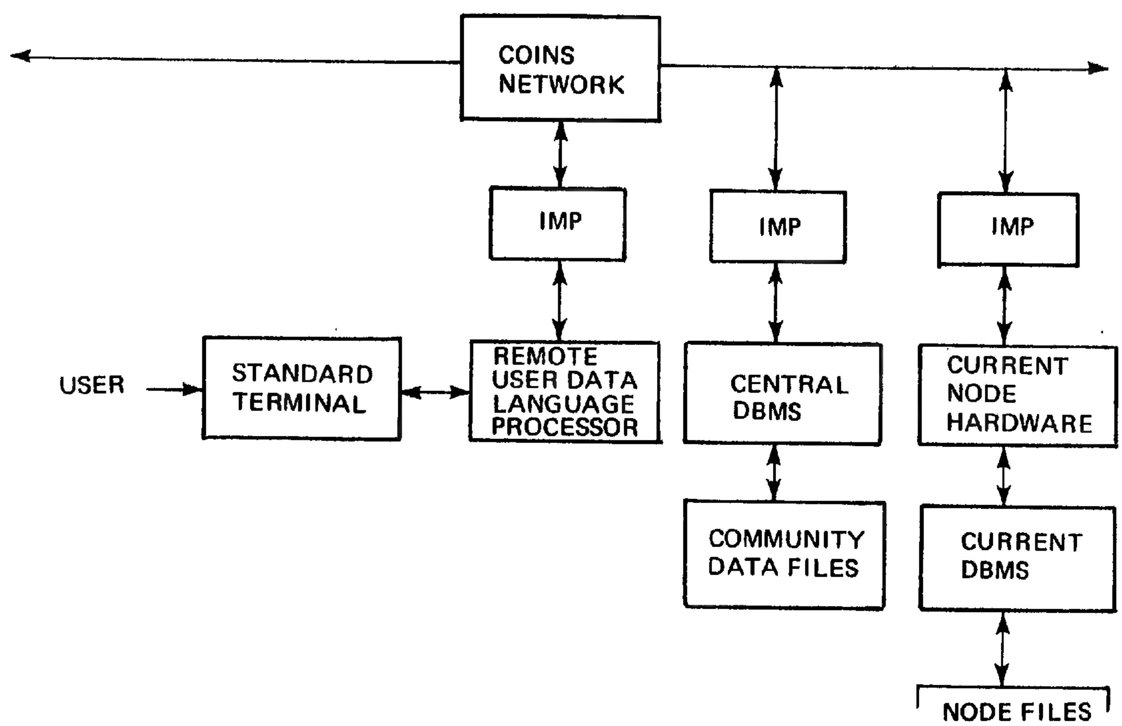

Figure 6-12. Basic Configuration for Solution 6

6.6.2 Solution 6 Operation

The operation of a COINS node using solution 6 is equivalent to the operation of solution 4 except that there are multiple user data language processors distributed to each node accessing a centralized DBMS node or nodes.

6.6.3 Solution 6 Tasks, Schedules, and Costs

The primary tasks required to implement solution 6 include:

a. Define the user data language for COINS.

b. Develop the user data language processors needed for each node.

c. Develop COINS centralized DBMS node(s).

d. Convert and install COINS files to be managed by central DBMS nodes.

e. Install user data language processors and central DBMS nodes.

f. Test each node's operation and all nodes with the network.

The schedules and costs associated with these tasks are shown in Figure 6-13.

6.6.4 Solution 6 Evaluation

An evaluation of the capabilities afforded to the COINS user, node, and network by solution 6 follows:

	Capabilities	Evaluation
1	User support	Solution 6, along with solution 7, provides the best level of user support for all types of COINS users.
2	Data language	As in solution 5, the data language requirements for COINS are completely satisfied by the Logicon uniform data language.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE USER DATA LANGUAGE						120
2. DEVELOP USER DATA LANGUAGE PROCESSOR (UDLP)						576
3. DEVELOP COINS CENTRAL DBMS						810
4. CONVERT AND INSTALL FILES						500
5. INSTALL UDLP AND CENTRAL DBMS						14,636
6. NODE AND NETWORK TEST						250
TOTAL						16,892

Figure 6-13. Schedules and Costs for Solution 6

	Capabilities	Evaluation
3	Data structures	Assuming the data structures provided by the centralized DBMS node are hierarchical or better, the data structure requirements for COINS are completely satisfied. The only possible problem arises if there is significant differences between the data supported by the centralized DBMS nodes and the file sponsor nodes. In this case file maintenance and transmission could be a problem.
4	Data management functions	The data management functions defined by the user data language are complete and uniform since these functions are implemented by the centralized DBMS nodes. The only exception which occurs involves the file sponsor's activities at the local nodes.

Capabilities		Evaluation
5	Uniform data management at node	Again, the centralized DBMS nodes provide uniform data management as seen by all COINS users, except the file sponsor.
6	Terminal compatibility	User data language processors at each node will provide standard terminals for COINS users.
7	Privacy and integrity	Excellent data privacy and integrity capabilities provided by solution 6 when duplicate file maintenance problem is excluded.
8	Network compatibility	Provides good network compatibility.
9	COINS schedule	Probable delay in full operation of COINS-II by 18 months.
10	Performance	Possible performance problem with centralized DBMS nodes, otherwise performance should be excellent.
11	Cost	Moderate cost, $16,942 per terminal when one thousand terminals served.

An evaluation of the potential problems associated with solution 6 afforded the following results.

Potential Problem		Evaluation
1	Potential performance problem due centralized DBMS.	Highly probable when the 8 to 1 update for intelligence files is considered.
2	File redundancy and updating by file sponsors.	Duplicate files will almost certainly exist and the current trend is for file sponsors to maintain files for their own agency most effectively.

6.6.5 Solution 6 Advantages/Disadvantages

Advantages		Disadvantages	
1	Single user data language	1	Possible performance problems at centralized DBMS nodes.
2	Little or no changes required at host nodes	2	File redundancy and update problems which oppose purpose of COINS.
3	Standard terminals and user interfaces	3	User data language processors and centralized DBMS nodes must be developed.
4	All COINS files under a single standard DBMS	4	Delay in COINS schedule is significant.

6.7 SOLUTION 7: CURRENT HARDWARE, REMOTE DBMS, AND UDLP

Solution 7 consists of using current node hardware and DBMS for nonshared data files and standard DBMS with dedicated processor (similar to the XDMS processor described in paragraph 5.5) for shared data files (i.e., shared by the COINS community). The shared data files are accessed via user data language processors.

6.7.1 Solution 7 Configuration

The basic configuration for solution 7 is shown in Figure 6-14. Two configuration alternatives exist for solution 7. The first alternative concerns the user data language processor and the free-standing IMP connected version is recommended. The second alternative involves interfacing the remote DBMS processor. Several options exist and they are:

a. Connect the remote DBMS processor to a data channel of the host node.

b. Connect the remote DBMS processor to the host node IMP.

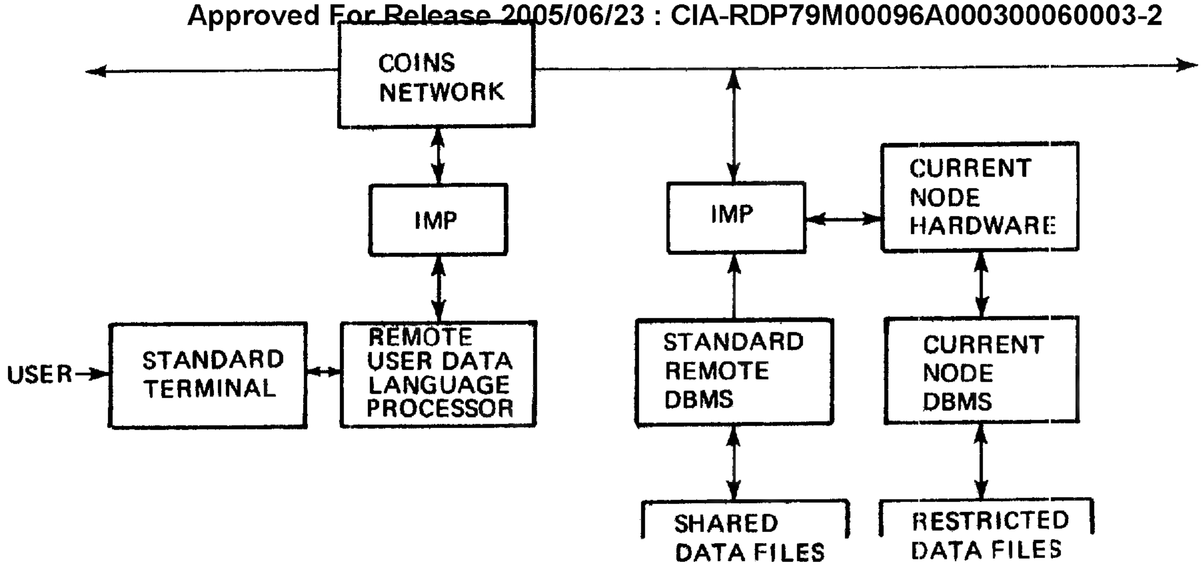

Figure 6-14. Solution 7 Configuration

c. Connect the remote DBMS processor to both the node system data channel and the node IMP.

Based upon an initial evaluation, connecting the remote DBMS processor to the host node via the IMP is recommended. The primary reason for this recommendation is due to the uncertainties involved at each host node data channel interface. In the case of the Univac 1100 series, the interface is relatively simple; while for the IBM 370, the data channel has been buried in the mass storage controller. And in the case of the Honeywell 6000 series, Honeywell has been unwilling to release the IOM and 355 specifications needed to evaluate the interfacing problem. As a result, the configuration recommended for solution 7 is as shown in Figure 6-14.

The user data language processors utilized for this solution are the same as described in paragraph 5.3. The remote DBMS processor proposed for this solution is equivalent to the XDMS processor described in paragraph 5. 5.

6.7.2 Solution 7 Operation

A COINS user enters his data transaction through the standard terminal supported by the user data language processor at his location. The user data language processor, after validating the user access rights, translates the transaction from external form of the COINS user data language into an internal form called the Data Interface Language (DIL) which is understood by the remote DBMS processor at each host node. The user's translated transaction is then sent to the appropriate remote DBMS processor for processing. When the remote DBMS processor has completed processing the transaction, the response to the transaction is sent back to the user's UDLP for formatting and distribution to the user.

6.7.3 Solution 7 Tasks, Schedules, and Costs

To implement solution 7, the following tasks are required:

a. Define the user data language to be supported for COINS and define the data interface language needed to interface with the remote DBMS processors.

b. Develop the user data language processors needed for each user node.

c. Develop the remote DBMS processor using the XDMS technology.

d. Develop the node interfaces needed for each target system to interface with the remote DBMS processors for the purpose of updating the shared files.

e. Install the user data language processors and remote DBMS processors at each node.

f. Test each node's operation and then the COINS-II network.

The schedules and costs associated with implementing solution 7 are shown in Figure 6-15.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE USER DATA AND DATA INTERFACE LANG						240
2. DEVELOP USER LANGUAGE PROCESSOR (UDLP)						576
3. DEVELOP STD DBMS PROCESSOR						576
4. DEVELOP HOST NODE INTERFACES						200
5. INSTALL UDLP AND STD DBMS						10,644
6. TEST NODES AND NETWORK						250
TOTAL						12,486

Figure 6-15. Solution 7 Schedules and Costs

6.7.4 Solution 7 Evaluation

An evaluation of the capabilities and potential problems associated with solution 7 with respect to the COINS user, node, and network requirements follows:

Capabilities		Evaluation
1	User support	Excellent user support available with solution 7 with highly interactive capabilities, online COINS user file guide support, command usage assistance, effective access control, and a standard user data language fully meeting COINS user language requirements.
2	Data language	The data language requirements are completely satisfied with a single user data language equivalent to the Logicon uniform data language.

	Capabilities	Evaluation
3	Data structures	The dedicated remote DBMS has the capability to provide any data structures needed by COINS users.
4	Data management functions	The data management functions provided by solution 7 in the areas of data definition, file creation, interrogation, updating, and display are unsurpassed by any other solution. Solution 6 has somewhat higher data manipulation due to processor capacity available at centralized DBMS.
5	Uniform data	Solution 7 provides the best approach to uniform data management at host node with the exception of solution 1. Solution 7 provides both logical and physical storage management and structure standardization.
6	Minimum impact at node	Solution 7 provides moderate impact due to the necessity to install and integrate both UDLPs and remote DBMs processors. Further, node interfaces must be developed and established transferring and updating files between the host node system and its associated remote DBMs processors.
7	Terminal compatibility	Standard terminals provided at all nodes.
8	Privacy and integrity	COINS data privacy and integrity are best satisfied by this solution. Two level access control is provided by UDLP and the remote DBMs processors. In addition, remote DBMS partitions total COINS data base thereby providing privacy and integrity. Finally, host node is a backup for refreshing remote DBMS in the event of failure, etc.
9	Network compatibility	Provides good network compatibility with single standard interface; IMP to UDLP, remote DBMS processor, and host node.
10	COINS schedule	Unfortunately, delays COINS schedule by over 18 months.

Capabilities	Evaluation
11 Performance	Solution 7 provides the best performance.
12 Cost	Only solutions 1, 3, 4, and 6 are more costly.

Potential problems which are associated with solution 7 are:

Potential Problems	Evaluation
1 Nodes may resist installation of remote DBMS processors because loss of files is feared	Highly likely and if this solution selected, this is the key issue which must be resolved. In fact, remote DBMS processors would belong to the nodes and be controlled by them, thereby providing more effective control for the nodes over their files.
2 Technical feasibility demonstrated with XDMS but not proven	Attaching the remote DBMS via the node IMP resolves many of the technical problems associated data channel interfaces. The primary issue to be resolved is bandwidth sufficiency of IMP-host node-remote DBMS data path.

6.7.5 Solution 7 Advantages/Disadvantages

Advantages	Disadvantages
1 Single user language	1 User data language and remote standard DBMS processors must be developed.
2 Standard DBMS at every host node in COINS	2 File conversions will be required to move files from host nodes to remote DBMS processor.
3 Standard terminals are available to all COINS users	3 Multiple node interface packages must be developed to enable file creation and maintenance by host nodes on their remote DBMS processors.
4 Extended user support is provided in terms of interaction, COINS user file guiles available online, and command usage support	4 Possible resistance due to "ownership and control of data files" when they are not contained explicitly in host node system but in remote DBMS processors.

6.8 SOLUTION 8: CURRENT SYSTEMS AND DATA SPECIALISTS

Solution 8 consists of retaining current hardware, DBMS, and multiple user data languages at the current and potential COINS nodes and establishing and training teams of data specialists to serve as interfaces to the COINS files and user languages for intelligence community users of COINS.

6.8.1 Solution 8 Configuration

The basic configuration for solution 8 is shown in Figure 6-17. The box labeled data specialist at nodes represents office areas with terminals where users wishing to access COINS files can come and request that data specialists assigned to that office either perform the queries on behalf of the user or assist the user in operating a terminal to perform his own data transactions.

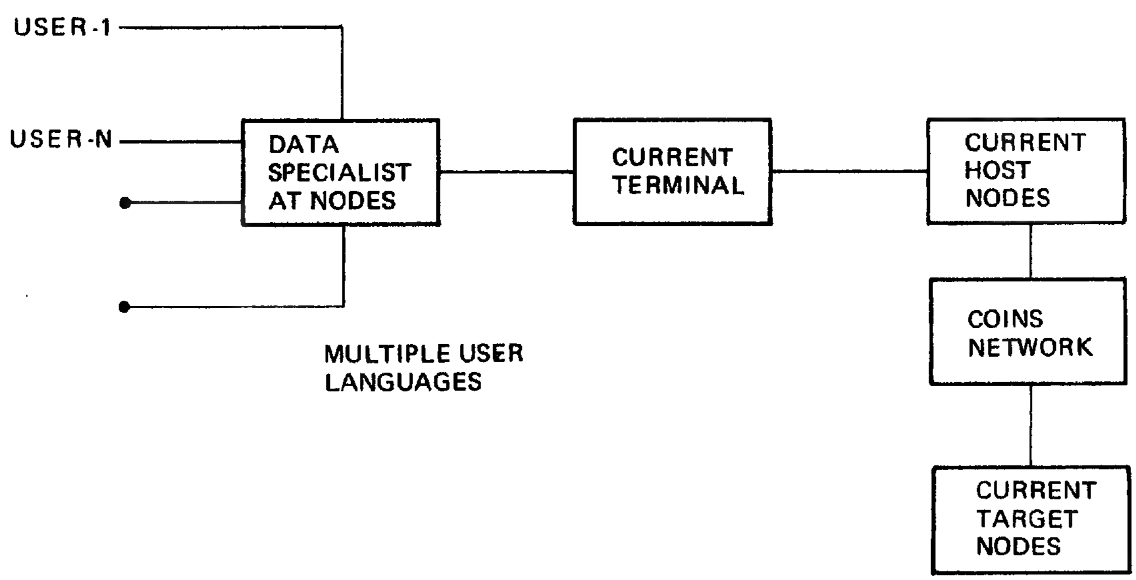

Figure 6-16. Basic Configuration for Solution 8

6.8.2 Solution 8 Operation

The operation of COINS does not change with solution 8. The only difference between current operations and solution 8 is the existence of personnel, i.e., data specialists, with specific training in all of the various query languages who are available to assist intelligence analysts in their use of COINS.

6.8.3 Solution 8 Tasks, Schedules, and Costs

The tasks which must be executed in order to implement solution 8 include:

a. Each agency must determine whether it wants to provide data specialists for its COINS users and if so, recruit the individuals needed for this work.

b. Each of the personnel selected for employment as data specialists must be trained in the use of each of the query languages currently being supported by COINS and any new languages as they are added. In addition, each data specialist must become very familiar with each file supported by COINS.

c. User service centers must be located and established.

The schedules and costs associated with performing these tasks are shown in Figure 6-17. The costs shown for solution 8 were calculated on the basis that all current host nodes would employ one data specialist for every 25 analysts using COINS. For example, with 200 analysts using COINS, 8 data specialists would be required on a 24-hour basis to support these analysts.

6.8.4 Solution 8 Evaluation

No detailed evaluation was performed for solution 8 because it is considered to be only a temporary solution to the COINS multi-language problem. Retreival technicians have proven useful at both the CIA and DIA. It is

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. RECRUIT DATA SPECIALISTS	—					144
2. TRAIN DATA SPECIALISTS	—					360
3. ESTABLISH USER SERVICE CENTERS	—					1,352
TOTAL						1,856

Figure 6-17. Schedules and Costs for Solution 8

unlikely on the other hand that retrieval technicians would be accepted or utilized at NSA or NPIC.

6.8.5 Solution 8 Advantages/Disadvantages

Advantages

1 The use of data specialists is the most rapid, although incomplete solution

2 Least costly of all the solutions

3 Excellent training for personnel wishing to specialize in intelligence data handling

Disadvantages

1 Personnel to staff the data specialists positions may not be available.

2 7 days per week, 24-hour duty cycle for data specialists will result in a periodic lack of coverage due to the lack of personnel.

3 High turnover rate likely among data specialists due to hours and availability of better assignments.

4 Perhaps the most important disadvantage is that this solution imposes human barriers between the intelligence analyst and his data.

6.9 SOLUTION 9: A COMBINATION OF SOLUTIONS 2 AND 5

Solution 9 is a combination of solutions 2 and 5 and consists of:

a. Initiating development of remote user data language processors for each node.

b. Selecting and installing a standard software DBMS at each node.

The tasks, schedules, and costs associated with this solution are essentially those shown for the independent solutions used to form this one with the exception that the user data language processor development time should be shorter due to the fact that translation and interfaces must be established to a single standard DBMS. The advantages and disadvantages of this solution are:

Advantages	Disadvantages
1 Single user language	1 User data language processor development required.
2 Standard DBMS	2 File and application conversion required.
3 Standard data definitions for all COINS files	3 Nonstandard physical storage structures.
4 Standard logical data structures with result that command semantics are identical at each node.	
5 Minimal hardware changes.	
6 Standard terminals.	
7 Extensive user support with user data language processors	

6.10 SOLUTION 10: A COMBINATION OF SOLUTIONS 2 AND 7

Solution 10 is a combination of solutions 2 and 7 and consists of:

a. Initiating development of remote user data language processors.

b. Selecting and installing standard software DBMS.

c. Initiating the development of remote standard DBMS processors.

The tasks, schedules, and costs associated with this solution are again essentially those shown for the independent solutions. Also the development cycle for the user data language process should be shorter due to the single interface required with the standard software DBMS. An assumption made for this solution is that the remote standard DBMS chosen for development will be consistent with the software DBMS selected for initial installation. The advantages and disadvantages of this solution are those shown for solution 7 with the added advantage of a shorter implementation cycle.

SECTION 7

EVALUATION OF POTENTIAL COINS SOLUTIONS

Each of the solutions described in Section 6 was evaluated against the requirements developed in Section 2. The evaluation was carried out using a modified form of the Kepner-Tregoe evaluation method. Essentially this evaluation method consists of assigning weights to each of the requirements indicated in Section 2. The range of values for these weights is from 1 to 10. Each solution is then examined for its capability to meet a given requirement. This evaluation is done on a requirement factor by requirement factor basis for each solution as opposed to evaluating each solution for all requirements at once. The purpose for evaluating on the requirement factor basis is to ensure solutions are only compared on a factor by factor basis thus providing a reasonable amount of objectivity. As each solution is evaluated with respect to a given factor, it is given a rating from 1 to 10 which denotes its capability to meet that requirement. When all the factors have been evaluated for all of the solutions, the ratings given to each solution for each requirement factor are multipled by the requirement factor weighting to produce a series of weight-rating products for each solution. These weight rating products are summed for each solution and this sum represents a quantitative evaluation of that solution with respect to the COINS requirements. The results of the capability evaluation for the potential COINS solution are shown in Figure 7-1. The upper half of each square represents the capability rating for a given solution, and the lower portion is the product of the rating times the requirement factor weighting. Subsequently, the potential problems associated with each solution were determined and a similar weighting-rating evaluation was done for each solution with respect to these potential problems. The potential problem evaluations are described in the remainder of this section.

REQUIREMENTS	WT	SOLN 1	SOLN 2	SOLN 3	SOLN 4	SOLN 5	SOLN 6	SOLN 7	SOLN 8	SOLN 9	SOLN 10
1. USER SUPPORT	5	7 / 35	6 / 30	7 / 35	8 / 40	9 / 45	8 / 40	9 / 45	5 / 25	9 / 45	10 / 50
2. DATA LANGUAGE	10	6 / 60	7 / 70	8 / 80	10 / 100	8 / 80	10 / 100	8 / 80	6 / 60	8 / 80	10 / 100
3. DATA STRUCTURES	7	7 / 49	8 / 56	6 / 42	8 / 56	6 / 42	8 / 56	10 / 70	7 / 49	8 / 56	10 / 70
4. DATA MANAGEMENT FUNCTIONS	10	8 / 80	8 / 80	6 / 60	8 / 80	7 / 70	8 / 80	10 / 100	6 / 60	8 / 80	10 / 100
5. UNIFORM DATA MGMT AT NODES	10	9 / 90	8 / 80	5 / 50	7 / 70	5 / 50	7 / 70	10 / 100	6 / 60	8 / 80	10 / 100
6. MINIMUM IMPACT ON NODES	7	1 / 7	7 / 49	9 / 63	6 / 42	7 / 49	10 / 70	8 / 56	5 / 35	8 / 56	7 / 49
7. TERMINAL COM-PATIBILITY	3	10 / 30	6 / 18	8 / 24	8 / 24	10 / 30	10 / 30	10 / 30	6 / 18	10 / 30	10 / 30
8. DATA PRIVACY AND INTEGRITY	10	5 / 50	6 / 60	7 / 70	8 / 80	7 / 70	8 / 80	10 / 100	5 / 50	6 / 60	10 / 100
9. NETWORK COM-PATIBILITY	6	10 / 60	8 / 48	7 / 42	8 / 48	8 / 48	8 / 48	8 / 48	7 / 42	9 / 54	8 / 48
10. COINS SCHEDULE	8	8 / 64	7 / 56	7 / 56	5 / 40	7 / 56	5 / 40	3 / 24	10 / 80	9 / 72	10 / 80
11. PERFORMANCE	10	7 / 70	7 / 70	4 / 40	5 / 50	8 / 80	6 / 60	8 / 80	2 / 20	9 / 90	10 / 100
12. COST	8	1 / 8	10 / 80	7 / 56	7 / 56	8 / 64	6 / 48	8 / 64	8 / 64	7 / 56	7 / 56
TOTALS		603	697	618	686	684	722	797	563	759	883

Figure 7-1. Potential COINS Solution Capability Evaluations

The potential problems associated with each solution and the evaluation of the impact of these problems follows:

Solution No. 1 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	GSA constraints against single vendor.	10	10	100
2.	Nodes constrained to single architecture and technology.	8	5	40
3.	Conversion to next generation equipment must be done in parallel.	7	10	70
4.	Probable delay in standardization at each node due to slow conversion.	5	10	50
5.	Any system selected as the standard will require extension to meet user requirements.	3	10	30
	Total			290

Solution No. 2 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	Failure of single software source to support standard DBMS.	10	2	20
2.	Maintenance and updating problems on multiple hardware types.	10	5	50
3.	Exclusion of certain hardware unless standard DBMS is developed for them.	2	10	20
4.	Any DBMS selected as standard will require modifications to meet user requirements.	4	10	40
	Total			130

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

Solution No. 3 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	Highly probable performance and reliability problems.	10	9	90
2.	Knowledge of node sublanguages needed to perform certain functions by file sponsors	10	4	40
3.	Uniform Terminal Process or approach not utilized.	10	2	20
	Total			150

Solution No. 4 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	High probable performance problems.	10	9	90
2.	Duplicate files required which will result in lack of support by owners.	10	9	90
3.	Vulnerability to damage of single center.	10	2	20
	Total			200

Solution No. 5 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	Delay due to software development required.	10	2	20
	Total			20

Solution No. 6 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	Highly probable performance problems.	10	9	90
2.	File redundancy and update problems.	9	8	72
3.	Reliability of single node.	10	3	30
	Total			192

Solution No. 7 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	Reluctance of nodes to put files on dedicated DBMS processor.	10	8	80
2.	Technological and system interface problems involving physical device/system protocols.	10	10	100
	Total			180

Solution No. 8 Potential Problem Analysis

	Problem	Wt	Prob	Eval
1.	Availability of retrieval technicians 24 hours/day for 7 days/week	7	8	56
2.	Psychological impedance to analyst.	10	10	100
3.	Lack of convenient access to retrieval technicians by analysts.	7	8	56
4.	Retrieval technician turnover due to attrition.	5	9	45
	Total			167

The potential problems for solutions 9 and 10 consist of at most the sum of the problems associated with each of the solutions used to form the mixed solutions. However, due to the simplification of user data language processor development through the use of a standard DBMS, the potential problems associated with solution 9 were considered equivalent to those of solution 2 and the potential problems associated with solution 10 were considered equivalent to those of solution 7.

Subtracting the potential problem ratings from the capability ratings for each solution produced a net evaluation for each solution. The results of this evaluation are shown in Figure 7-2. An overall summary of the evaluation of the potential COINS solutions is given by Figure 7-3.

SOLUTION	CAPABILITY EVALUATION	PROBLEM EVALUATION	NET VALUE OF SOLUTION	RANK
1	603	290	313	10
2	697	130	567	5
3	618	150	468	8
4	686	200	486	7
5	684	20	664	2
6	722	192	530	6
7	797	180	617	4
8	563	167	396	9
9	759	130	629	3
10	883	180	703	1

Figure 7-2. Results of Evaluation

SOLUTIONS	SCHEDULE MONTHS	EFFORT MAN YEARS	HARDWARE COST/K$	SOFTWARE COST/K$	TOTAL COST/K$	RANK
1. STANDARD HARDWARE, DBMS, AND UDLP	33	154	14,427	8,366	22,793	10
2. SELECTED HARDWARE, STANDARD DBMS AND UDLP	36	81	–	3,656	3,656	5
3. CURRENT HARDWARE AND DBMS, CENTRAL UDLP	36	25	12,870	1,226	14,096	8
4. CURRENT HARDWARE, CENTRAL DBMS, AND UDLP	42	53	12,980	2,768	15,748	7
5. CURRENT HARDWARE AND DBMS, REMOTE UDLP	36	29	8,516	1,396	9,912	2
6. CURRENT HARDWARE, CENTRAL DBMS, REMOTE UDLP	42	48	14,470	2,472	16,942	6
7. CURRENT HARDWARE, REMOTE DBMS, AND UDLP	54	57	9,924	2,562	12,486	4
8. DATA LIBRARY SPECIALISTS	12	44	–	1,856	1,856	9
9. SOLUTIONS 2 AND 5	24, 36	110	8,516	5,052	13,568	3
10. SOLUTIONS 2 AND 7	18, 24, 42	112	9,216	5,644	14,860	1

Figure 7-3. Solution Evaluation Summary

SECTION 8

DESCRIPTION OF RECOMMENDED APPROACH

Based upon the results of the evaluation described in the previous section Logicon recommends:

1. Solution 10 be implemented to solve the COINS multi-language problem.

2. Solution 10 be implemented in the following evolutionary phases.

   a. Construct and install Uniform Terminal Processors (UTP) with IMPS at each agency supported by COINS to enable a uniform interface between each users terminal and COINS, plus providing online command and file usage support for the languages and files currently supported by COINS.

   b. Select and install a standard software DBMS at each node supplying files to the intelligence community through COINS thus establishing standardized data structure and accessing interfaces for COINS with these files.

   c. Concurrent with phase "b", develop and install a Uniform Data Language Processor (UDLP) on each Uniform Terminal Processor system to translate user requests from the COINS Uniform Data Language into the command language of the standard DBMS thereby providing each COINS user with a uniform interface to all current and future COINS files.

   d. Initiate development of dedicated processors to enable direct hardware implementation of the standard DBMS chosen for COINS and to provide the storage capacity, access performance, and data security needed for additional files and users.

3. The IMPS, Uniform Terminal Processors, Uniform Data Language Processors, and dedicated DBMS processors required for this solution should all be considered part of the COINS network; and therefore should be produced, installed, and managed by the COINS PMO on behalf of each agency to insure effective installation and uniform operation of these facilities for the intelligence community.

The purpose, tasks, schedules, costs, and results associated with each phase of the recommended approach are described below.

8.1 UNIFORM TERMINAL PROCESSOR CONSTRUCTION

The purpose of this activity is to construct and install at each agency served by COINS, one or more Uniform Terminal Processor systems. The UTP provides the users of COINS II with a uniform logical interface through which to enter his requests, interactive online user support in command and file guide usage, and serve as a base system upon which to install the COINS Uniform Data Language Processor.

Each Uniform Terminal Processor (UTP) system consists of an IMP, an asynchronous communications multiplexer, two mini-computer processors (PDP-11/45 or PDP-11/70), a mass-storage controller and devices, 16 - 32 asynchronous communication lines and adapters to interface the user agency's terminals and IMP-UTP communications interface. In addition, synchronous communications facilities may be desired between the UTP and certain host node systems. Figure 8-1 depicts a typical configuration for a Uniform Terminal Processor system. We recommend that community terminal standards be established specifying a selected set of terminal types to be used within the community. The construction and operation of the UTP systems can be enhanced if terminal standards are established at least on an agency by agency basis.

The tasks, schedules, and costs associated with performing this activity are shown in Figure 8-2. Completion of this activity will result in a configuration similar to that shown in Figure 8-3 to exist at each agency served by COINS.

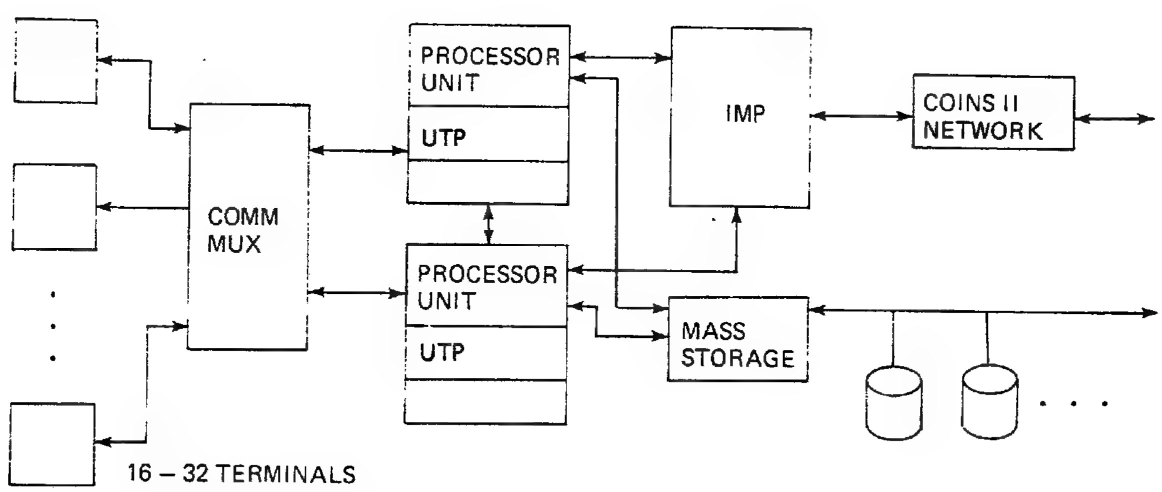

Figure 8-1. Typical UTP Configuration

TASKS	SCHEDULE 76	77	78	79	80	COSTS/K$
1. SPECIFY UTP AND STANDARD TERMINALS	—					36
2. DEVELOP BASIC UTP SYSTEM	—					144
3. DEVELOP USER AND FILE SUPPORT	—					48
4. TEST UTP AT NODE		—				48
5. TEST UTPS ON NETWORK		—				48
TOTAL						324

Figure 8-2. Uniform Terminal Processor Construction Tasks, Schedules, Costs

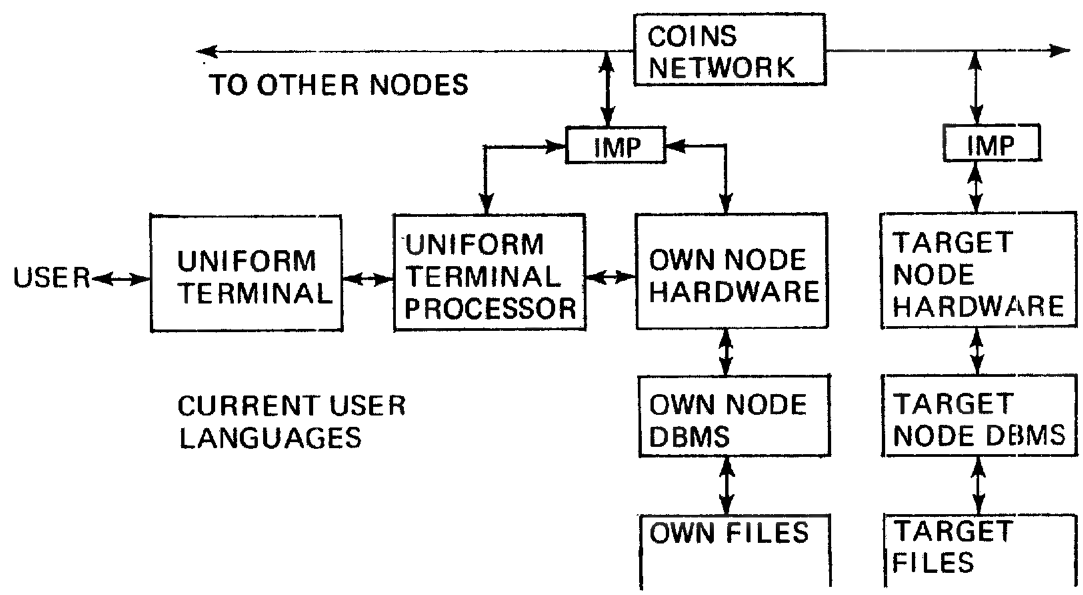

Figure 8-3. Uniform Terminal Processor Configurations at Node

8.2 STANDARD DBMS SELECTION AND INSTALLATION

The purpose of this activity is to select and install at each host node a standard software DBMS (host nodes are those nodes who are providing files for access by COINS users). A standard DBMS at each host node will provide a single logical interface between the COINS uniform data language and the host node and a single logical interface between the host node and its data bases. In addition, the data descriptions for each COINS file will become more uniform. Further, a standard DBMS at each node provides a standard interface for application programs and opens up the possibility of application program sharing in addition to a data sharing. Finally, the installation of a standard DBMS at each node lays the foundation needed for developing remote DBMS processors using dedicated hardware such as the Bell Telephone Laboratories processor. The tasks, schedules, and costs associated with performing this activity are shown in Figure 8-4.

Due to the diversity of the hardware currently being used at the various current and potential host nodes in COINS, the MRI System 2000 should

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. REFINE STANDARD DBMS REQUIREMENTS	—					30
2. EVALUATE AND SELECT STANDARD DBMS	—					30
3. INSTALL STD DBMS AT NPIC WITH FILES	—	—				150
4. INSTALL STD DBMS AT NSA WITH FILES		—				1600
5. INSTALL STD DBMS AT DIA WITH FILES		—				1210
TOTAL						3020

Figure 8-4. Tasks, Schedules, and Costs for Standard DBMS Installation

probably be chosen as the standard DBMS. However, if NSA were to select either IBM or Univac equipment for TILE and perhaps SOLIS and DIA were to select similar equipment for the new DIAOLS hardware, the CODASYL DBTG DBMS available from Univac and Cullinane should be selected as the standard DBMS. NPIC has already selected the Univac 1110 and DMS-11. The result of installing a standard DBMS at each of the host nodes in COINS would result in a configuration similar to that shown in Figure 8-5 for each node. Note that the current DBMS at each host node would still be used until the COINS Uniform Data Language Processors were installed.

8.3 UNIFORM DATA LANGUAGE PROCESSOR CONSTRUCTION

The purpose of this activity is to provide COINS users with a standard user data language which will allow him to access any of the files supported by COINS. The Logicon Uniform Data Language is recommended for adoption as the COINS standard user data language. The tasks, schedules, and costs associated with this activity are shown in Figure 8-6. Note that within two years, the initial version of the Uniform Data Language is available to COINS users. This version allows complete interrogation, manipulation, updating, and display capabilities to the user for accessing COINS files managed by the standard DBMS selected for COINS host nodes and all of these functions are accomplished by the user at his terminal. The final version of the Uniform Data Language will provide capabilities for data definition, file creation, distributed data base searches, event triggered message generation, automatic routing of user data transactions (queries, updates, etc.), and specialized report generation. Addition of the Uniform Data Language Processor to the UTP is shown in Figure 8-7. The result of completing this activity is shown in Figure 8-8 which illustrates the configuration of a typical node. Due to the possibility of delays in installing standard DBMS at each host node, the Uniform Data Language Processor has been designed to translate the Uniform Data Language to multiple DBMS command languages. Therefore, should any host node be delayed with its standard DBMS, COINS users can still access all COINS files with a Uniform Data Language.

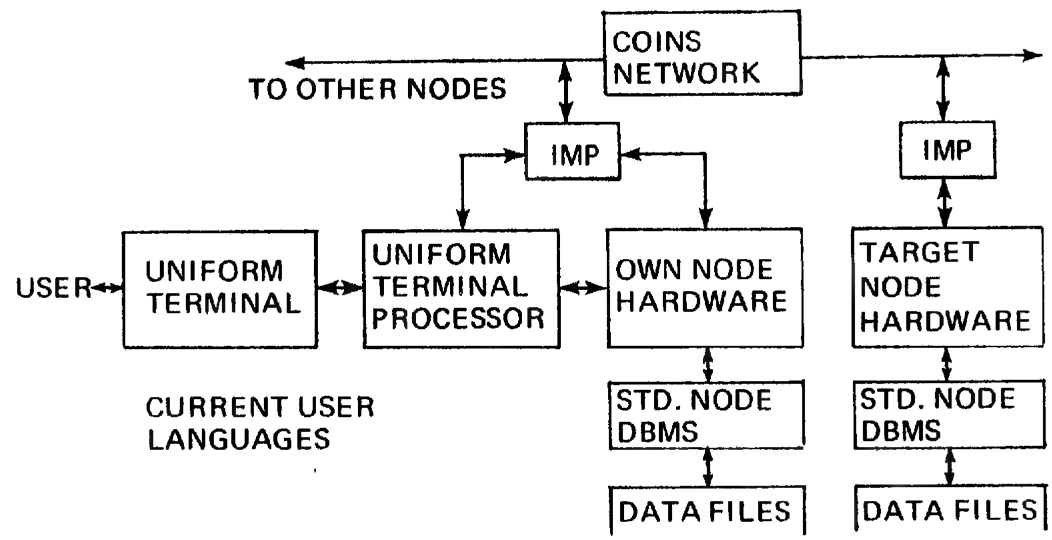

Figure 8-5. Configuration of Typical Node with Standard DBMS

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE UNIFORM DATA LANGUAGE						120
2. DEVELOP INITIAL UDLP						300
3. UPGRADE UTP TO UDLP AT DIA, NPIC, AND NSA						96
4. DEVELOP FINAL UDLP						240
5. INTEGRATE AND INSTALL AT ALL NODES						144
TOTAL						900

Figure 8-6. Tasks, Schedule, and Costs for UDLP Construction

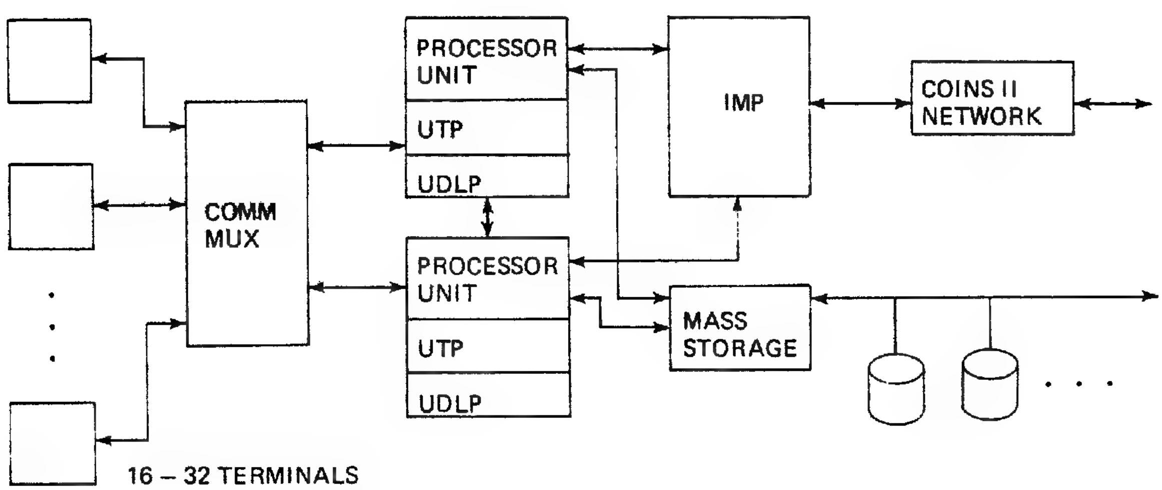

Figure 8-7. Typical UTP Configuration with Uniform Data Language Processor.

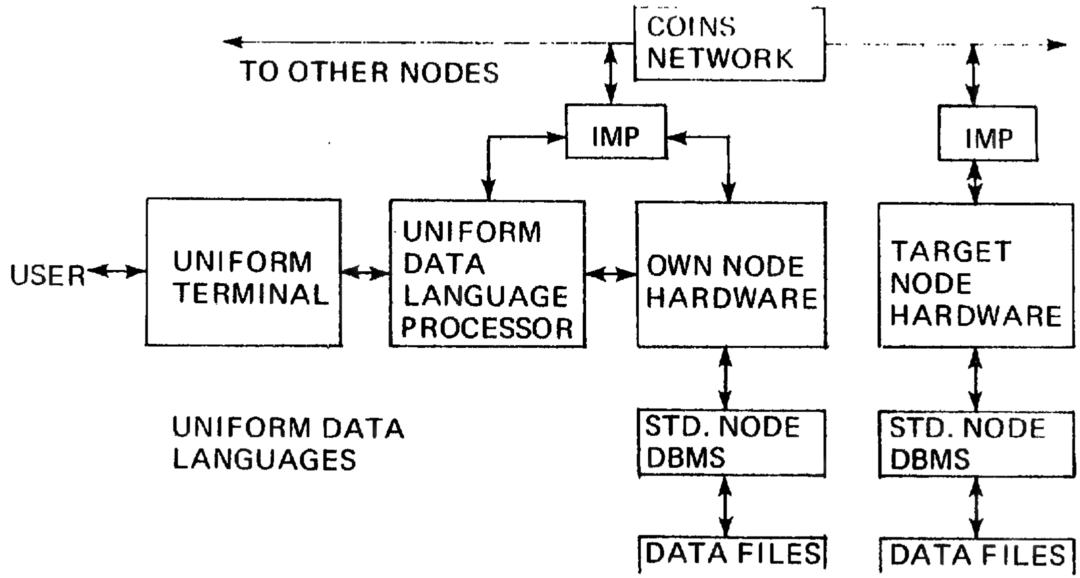

Figure 8-8. Typical Configuration of Node with UDLP and Standard DBMS

8.4 DEDICATED DBMS PROCESSOR DEVELOPMENT

The purpose of this activity is to provide COINS users and file sponsors with a high performance version of the standard DBMS selected previously using a dedicated processor. The overall functional organization of the dedicated DBMS processor is shown in Figure 8-9. The processor executive provides for overall control of the activities of the DBMS processor. The comm and trans mgr is responsible for communications control and transaction management between the processor and the COINS network or the host system. The command processor accepts commands from the transaction manager and selects the appropriate processor for activation based upon the type of command; i. e., data definition, file creation, interrogation, etc. The device strategy and mapping modules along with the device control modules are responsible to the actual accessing and operation (read, write, select, etc.) of the mass storage devices attached to the dedicated DBMS processor.

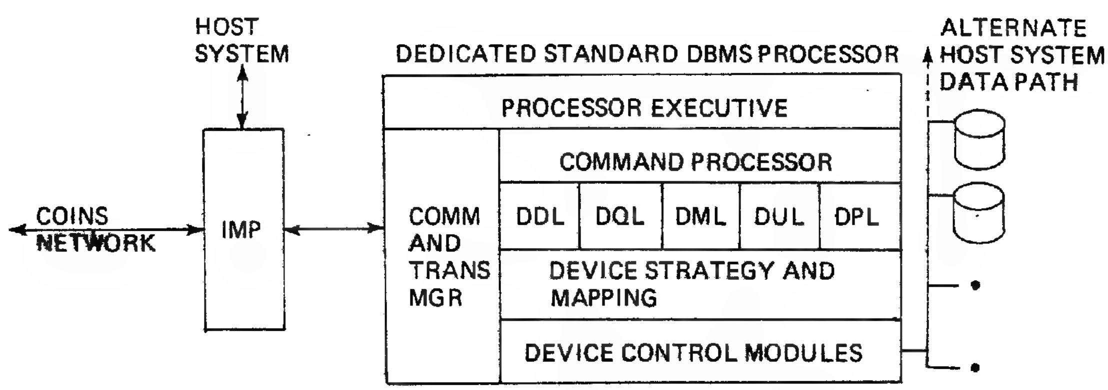

Figure 8-9. Functional Organization of Dedicated DBMS Processor

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2

The tasks, schedules, and costs associated with these activities are shown in Figure 8-10.

The result of completing this activity are shown for a typical node in Figure 8-11. Note that from a logical standpoint, the configuration in Figure 8-8 and Figure 8-11 are the same.

TASKS	SCHEDULE					COSTS/K$
	76	77	78	79	80	
1. DEFINE DEDICATED DBMS REQUIREMENTS						100
2. EVALUATE AND SELECT PROCESSOR AND STORAGE						100
3. DESIGN AND IMPLEMENT PROTOTYPE DBMS PROCESSOR						450
4. INSTALL AND TEST ON SELECTED NODE						100
5. CONSTRUCT PRODUCTION DBMS PROCESSORS						450
6. INSTALL AT EACH FILE SHARING NODE						200
SOFTWARE						1,400
HARDWARE (FOR EIGHT NODES)						2,816
TOTAL						4,216

Figure 8-10. Dedicated DBMS Processor Development, Tasks, Schedules, and Costs

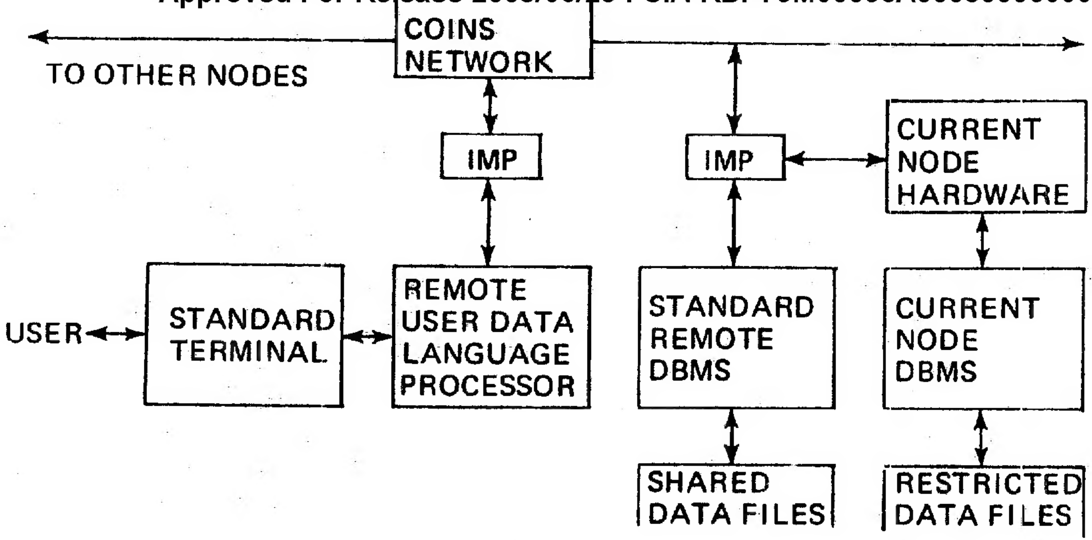

Figure 8-11. Typical Configuration for Node with Dedicated DBMS Processor

8. 5 RECOMMENDED APPROACH SUMMARY

In summary, Logicon recommends that the approach described in this section be implemented immediately. The advantages of this approach are:

a. A user interface is provided to COINS users within 18 months which allows these users to access COINS II in a uniform manner at all agencies. In addition, significant user support is provided in terms of command and file usage currently available.

b. Each user will be able to access all files for which he has access in a single Uniform Data Language which has been specifically constructed for the COINS community. This language will be available to all COINS users within two years and an extended version of this language will be available in three years.

c. Each host node will have a standard DBMS and as a result not only will files be more readily sharable in COINS but throughout the whole intelligence community.

d. The development of hardware versions of the standard DBMS will allow new data storage technologies to be utilized and provide for significant improvements in data base management performance.

e. This approach provides near term solution of the COINS multi-language problem and further, it is evolutionary so that a long term solution is also provided.

f. This solution provides for standard hardware and software at all user/system interfaces and at all system/data base interfaces while maintaining the capabilities provided by current node hardware and software.

The disadvantages of this approach are:

a. The need to rely upon a single DBMS vendor. This can be offset by the willingness of the COINS nodes to take responsibility for development and maintenance of the DBMS. In view of the nature of the data being managed by DBMS, this action is probably essential anyway.

b. The risk of the development efforts required for the Uniform Terminal Processor and the Uniform Data Language Processor. Because of the design and feasibility studies already completed, the major risk is primarily in the area of production management as opposed to technical development. Production management problems can almost always be resolved.

c. Each node will be required to convert their files and application programs to the new standard DBMS. This is the most significant disadvantage of this approach but experience has shown that files can be converted in about two weeks to one month at a total cost of less than $10, 000 per file.

A chart summarizing the elapsed time, man years of effort required, hardware costs, software costs, total costs, and the results obtained for this approach are shown in Figure 8-12. A detailed functional specification for the UTP, UDLP, and standard DBMS appear in Volume III of the final report for this study.

PHASE	ELAPSED TIME	MAN YEARS	HARDWARE COST/K$	SOFTWARE COST/K$	TOTAL COST/K$	RESULTS
1. UTP CONSTRUCTION	18 MO.	7	3,600	324	3,924	USERS NOW INTERFACING IN UNIFORM WAY WITH AUTO USER HELP AND FILE GUIDES
2. STANDARD DBMS	24 MO.	60	–	3,020	3,020	STANDARD LOGICAL STRUCTURE AND INTERFACES TO COMMUNITY DATA BASES
3. UDLP CONSTRUCTION	30 MO.	17	2,800	900	3,700	UNIFORM USER ACCESS TO STANDARD COMMUNITY DATA BASES
4. DBMS PROCESSOR DEVELOPMENT	48 MO.	28	2,816	1,400	4,216	PROVIDES PERMANENT SOLUTION TO STANDARD DBMS REQUIREMENT
TOTALS			9,216	5,644	14,860	

Figure 8-12. Recommended Approach Summary

STAT

Next 3 Page(s) In Document Exempt

Approved For Release 2005/06/23 : CIA-RDP79M00096A000300060003-2